HOST LINKS

GLINE™

Line Handlers and Configuration

http://www.gar.no/hostlinks/

G&R
GALLAGHER
ROBERTSON

Microsoft, Windows, MS, MS-DOS are registered trademarks of Microsoft Corp.
IBM and PC are registered trademarks of IBM Corp.
UNIX is a registered trademark in the United States and other countries, licensed exclusively through X/Open Company, Ltd.

Any other product names are trademarks of their respective owners.

Version 6.1

© Gallagher & Robertson as 1990-2004
All Rights Reserved

GALLAGHER & ROBERTSON AS, Kongens gate 23, N- 0153 Oslo, Norway
Tel: +47 23357800 • Fax: +47 23357801
www: http://www.gar.no/hostlinks/
e-mail: hostlink@gar.no

Contents

Installation **1**

Scope of the product **3**

Overview **5**
G&R product architecture 5
Architecture diagram 6
Line handlers 6
Handler/product relationships 7
Enclosures and ‘turn’ 9

Host Links Trace **11**
Trace activation 11
Trace types 11
Structure 12
Tracing Ggate 13
Examples - G&R products 13
CPI-C and Gweb trace files 15

Line handlers **17**
The DSA/DIWS handler 17
DSA/DIWS on Windows servers 18
Transport stacks 19
Accepting connections 21
The listener 21
Listener parameter overview 22
Using the Windows listener 23
Using UNIX/Linux listeners 23
Mailbox pools for incoming connects 24
Starting applications on demand 24
Issuing connections 26
Service messages 27
The connect command 28
Connecting to CXI applications 29
The disconnect command 29
The break commands 29
The identification inquiry 30

Connection errors 30
Parameters for DSA/DIWS handler 31
DSA and DIWS parameters in detail 36
The TCP/IP handler 47
Making connections using TCP 47
Accepting a connect 47
Issuing a connect 47
Service messages 48
The connect command 48
The disconnect and break commands 49
Parameters for the TCP/IP handler 49
TCP parameters in detail 50
The X.25 handler 57
Making connections using X25 57
Accepting a connect 57
Issuing a connect 57
Service messages 57
The connect command 58
The disconnect and break commands 58
Parameters for the X.25 handler 59
The Asynchronous (TTY) handler 61
Making connections using TTY 61
Accepting a connect 61
Issuing a connect 61
Service messages 62
The connect command 62
The disconnect and break commands 62
Parameters for the TTY handler 63

DSA Configuration 65
Step-By-Step Guide 65
Configuration check list 65
The mainframe system 65
The Host Links system 66
Configuring the mainframe 66
GCOS8 configuration 66
GCOS7 configuration 66
Front-end configuration 67
GCOS6 configuration 67
Another Host Links system 68
Configuring the Host Links system 69
Configuring the transport stack 69
Configuring Host Links 70

The dsa.cfg configuration file ... 74
Overview ... 74
Configuration file directives ... 76
SC - Session control ... 76
RSC - Remote Session Control ... 77
TS - Transport Station ... 78
TP - Transport Provider ... 80
CONAME - Connection Name ... 82
Gline parameters ... 83
PARAMS - Parameters ... 83
POOL - Parameter Pool ... 84
Macro facility ... 85
COSTRICT - Connection Name Strictness ... 85
RESTRICT - Parameter Restrictions ... 86
FILTER - Access Filter ... 86
Use of filters to restrict access ... 88
Sample dsa.cfg for RFC1006 ... 90
Brief description of the configuration ... 90
The Host Links node ‘HL01’ ... 91
The Host Links node ‘HL02’ ... 91
The DPS9000 node ‘GC8F’ ... 92
The DPS7000 node ‘GC7F’ ... 92
The MainWay ... 93
Sample dsa.cfg for OSI Transport ... 94
Sample configuration ... 94
Brief description of the configuration ... 95
The configuration files ... 96
The Host Links node ‘GRDL’ ... 96
The Host Links node ‘GRIL’ ... 96
The Host Links node ‘GRDW’ ... 97
The Host Links node ‘GRIW’ ... 98
The DPS6 node ‘B6Dx’ ... 98
The DPS7 node ‘B7DL’ ... 100
The DPS9000 node ‘B8DF’ ... 100
The Datanet node ‘DN01’ ... 101
The DPS7 node ‘B7DC’ ... 102
The DPS7 node ‘B7DF’ ... 102
A shared dsa.cfg for all the Host Links nodes ... 103

Appendix: Host Links Manuals 107

Appendix: DSA test utilities 109

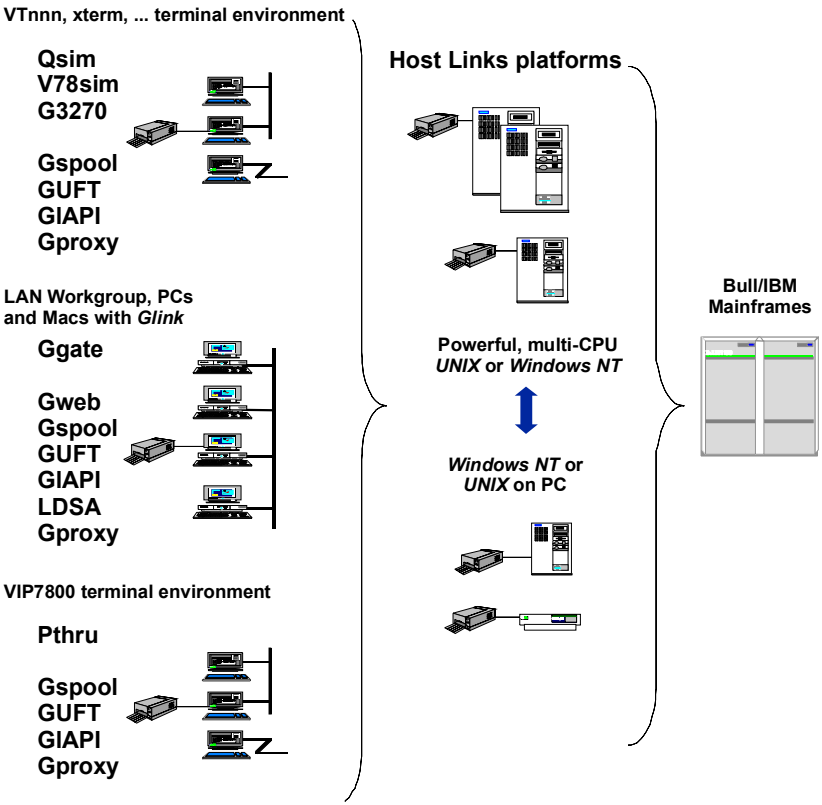
Gerror 109
Glnode 109
Gmacfix 110
Gping 110
Grnode 110
Gtrace 111
Gtsupd 111

Appendix: Error Codes 113

OSI/DSA error codes 113
Windows Sockets error Codes 125

Installation

The G&R emulations and gateways are independent programs, but part of the `G&R Host Links` product set available on all major UNIX/Linux platforms. Many of the products are also available for Windows servers. For details on platforms supported, software delivery and installation refer to the *Host Links Installation and Configuration* manual.



Scope of the product

G&R/Gline comprises a set of communications line handlers with a standard interface used by the *G&R Host Links* product set. The Gline line handlers standardize handling of any kind of communications line. This reduces application complexity, and makes applications portable across network environments. Applications are written using one of the communications libraries supporting the standard communications interface, *GlAPI* (Gline Application Programming Interface). The interface works with any Gline line handler, and can be used to program applications for communication over any network connection for which there is a Gline line handler, including:

Asynchronous	direct or modem connected
X.25	raw or PAD
TCP/IP	dial-up, ISDN, leased line, raw, Telnet, TNVIP TN5250 and TN3270
DSA/OSI	native Bull DSA and DSA/ISO Work Station over OSI-transport
DSA/RFC1006	native Bull DSA session over TCP/IP transport

GlAPI is available on all the supported platforms: UNIX/Linux and Windows servers and workstations.

The *G&R Host Links* product set uses the native mode *Gline API* for all communications.

Third party software houses and customers also use GlAPI when developing communications applications. GlAPI includes CPI-C for use in applications following the X/Open standard for programmatic access to communications. For details of the various APIs please refer to the *GlAPI Application Programming Interfaces* manual.

Overview

G&R product architecture

The G&R products are built in a modular way, and designed to take advantage of the multitasking capability of the Windows and UNIX/Linux platforms. Thus a single instance of a product will in general consist of two or three quite separate processes:

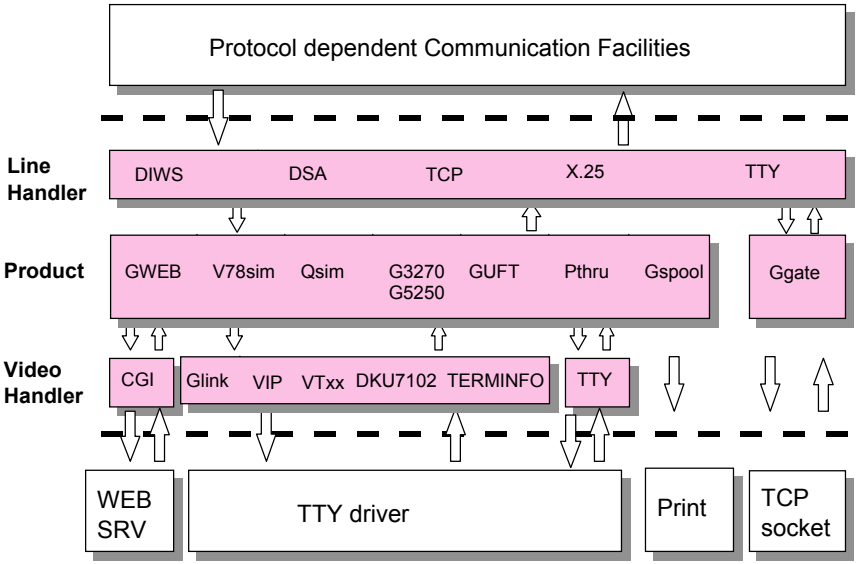
- A video handler: which maps all updates of the internal screen image into the control sequences required to update the specific screen being used. The video handlers are used by UNIX/Linux products, which must communicate with various different terminal types. The Windows products use the standard Windows graphical interface.
- A product: for example an emulator, which updates the internal screen image to duplicate the terminal being emulated. The product will also in general send and receive data to the host system using the global interface, which applies to all communications protocols.
- A line handler: which maps the global communications interface into the specific line protocol being used.

This structure has proven itself extremely efficient and very robust. We are able to develop a video handler for a new screen type, and we know that once it works with one G&R product it will work with all. We are able to develop a new line handler and know that once it works with one product it will work with all. We are able to develop a new product using a given screen and communications protocol, and know that the product will work with all screen types we support, and with all the communications protocols we support.

The server programs (e.g. *Ggate, Gspool, GUFT*) differ slightly from the above model, as they don't need a video handler, only a line handler. *Pthru* doesn't need a video handler either; it needs one line handler to connect to the host and one line handler (TTY) to pass all presentation data through to the terminal.

Architecture diagram

Clients for terminal users (*V78sim*, *Qsim*, *G3270*, G5250 *Pthru*) on UNIX and Linux only.



Line handlers

A *Host Links* product will generally start up the line handler automatically, and will in general itself accept parameters for tailoring the line handler, and pass these to the handler after contact is established.

We currently have the following line handlers: gl_dsa, gl_tcp, gl_x25 and gl_tty, handling DSA/DIWS, TCP/IP, X.25 and Asynchronous communication respectively (gl_x25 and gl_tty are currently only available for UNIX/Linux). Most of the *Host Links* products use one line handler at a time, but *Pthru* is an exception as it uses two line handlers, normally gl_dsa on the mainframe side and gl_tty on the terminal side.

Handler/product relationships

Any product using the *Gline API* to communicate with the line handler can be run using any line handler. This is one of the strengths of the architecture.

There are restrictions enforced by the external environment. For example the Questar DKU emulation *Qsim* can normally only be used over a link that the Bull primary network regards as synchronous. In practice this means it can only be run over a DSA/DIWS or TNVIP link.

The following is a list of the all the *Host Links* products and the line handlers they normally use on the host side:

Qsim	gl_dsa, gl_tcp
V78sim	gl_dsa, gl_tcp, gl_tty
G3270	gl_dsa, gl_tcp, gl_x25
G5250	gl_tcp
Pthru	gl_dsa, gl_tcp, gl_tty
Ggate	gl_dsa
Gspool	gl_dsa, gl_tcp
GUFT	gl_dsa
GlAPI	gl_dsa, gl_tcp, gl_x25, gl_tty

DSA connections can be made in the traditional way using OSI-transport, which is a requirement when connecting via old-style Datanets. The Bull systems can be accessed over an X.25 WAN or Ethernet LAN through a Datanet or MainWay front-end. Alternatively access can be direct to GCOS6 using a LAN adapter or direct to GCOS7 using ISL. Access can be by an FDDI LAN direct to GCOS7 using FCP7 or direct to GCOS8 using FCP8 (both support OSI-transport).

DSA connections can also be made over a TCP/IP network, using the Internet standard RFC1006 transport protocol to replace OSI-transport. MainWay front-ends with an ONP (Open Network Processor) have RFC1006 support in the standard product, allowing DSA sessions over TCP/IP into the MainWay. RFC1006 can also be installed in the FCP7 and FCP8 cards to support DSA connections direct to the mainframes without passing through the front-end. The GNSP of newer GCOS8 systems, and the Open System personality of GCOS7 Diane systems both support RFC1006 connections.

The *G&R/Ggate* product off-loads the DSA session protocol into gateways. By running Ggate on the system(s) with the mainframe connections the emulator platforms need only the very small and efficient Ggate protocol layer to connect over TCP/IP to a Ggate gateway with full native Bull network functionality. Ggate can make the mainframe connection using OSI-transport or RFC1006. If you must use OSI-transport for the mainframe connection, using Ggate will limit the need for OSI-stacks to the Ggate platforms.

The ***Host Links Installation and Configuration*** manuals cover the OSI stack installation and setup for the supported UNIX and Windows server systems.

Bull systems can also be accessed using Telnet VIP (TNVIP). The TNVIP servers in the MainWay, the Bull DPX/20, GCOS8 GNSP, the Open Systems personalities of GCOS7 Diane systems and *G&R/Ggate* are all qualified. However, RFC1006 will increase throughput as compared to using TNVIP. It will also give a real, fully functional DSA or DIWS session over the TCP/IP network, as compared to the limited terminal session offered by TNVIP.

IBM systems can be accessed using DSA into the Bull front-end and the OSF/SNA gateway (supported in the Datanet and MainWay). IBM systems can also be accessed using Telnet 3270 (TN3270 or TN3270E) to connect to any TN3270⇔SNA gateway or front-end. The MainWay gateway, the TN3270 server on the Bull DPX/20 UNIX systems, the IBM TN3270 front-end and the TN3270 server for Windows are all qualified. Additionally, IBM AS400 systems, or other host systems that supports the 5250 family of terminals, can be accessed using Telnet 5250 (TN5250) connecting through a Telnet 5250 server.

Enclosures and ‘turn’

The interface between a *Host Links* product and the line handler is global throughout the G&R product range, and for all user applications using one of the *GlAPI* programmatic interfaces for communication.

The functions for receiving and sending data include an indication of the ‘enclosure’ level for the data received, or to be sent. Enclosures are very important in the more advanced line handlers, particularly the Bull DSA handler. Enclosures are supported for all line handlers to make the API global and to make applications using the API independent of the type of communications line. If the enclosure level signaling is unreliable the product using the line handler will need special parameters to help it 'guess' when mainframe output is finished and it is able to process the data received.

The following enclosures are defined:

none
the enclosure is used for convenience of delivery to or from a lower layer of protocol and has no significance for the product using the interface.

end of segment
This is a complete session protocol unit, but is treated as ‘none’ above.

end of message
Defines a complete logical block of data transmitted or received, but indicates that more logical blocks will follow. This is the Bull DSA 'End of Quarantine Unit'.

end of group
Used by interactive products to indicate that the sender will now wait for input before continuing. This is known as the ‘turn’ concept.

Using Bull DSA terminology; when transmitting or receiving data, an application record, called a ‘message’, can be split (partitioned) into several DSA session protocol units called ‘letters’. The letter can be segmented due to size constraints at the session level (the maximum letter size). The session protocol units might have to be split again into several transport protocol units called ‘fragments’ for transmission due to size constraints at the transport level. Finally the transport units might be split again into ‘packets’ because of constraints at the network level.

Whereas a simple handler might have no session protocol unit size restriction it might have a transport size restriction, or a network packet size restriction (e.g. TCP/IP packet size) that would cause fragmentation of the application record for transmission purposes. Depending on the interface between the handler and the lower level protocol this might cause the receiver to see a series of data blocks with enclosure 'none'.

In two-way-alternate sessions 'end of group' is used to signal that a reply is now required (turn). The DSA record header indicates enclosure and turn, and in the DSA/ISO Work station protocol (DIWS) this is mapped to the OSI data token that allows the holder to transmit. There is really no turn concept in a two-way-simultaneous dialog, and no data token in two-way-simultaneous OSI sessions. However, the DSA records of the DSA/DIWS handler can still use 'end of group' enclosures and indicate turn to the Bull mainframe applications, which although they might support two-way-simultaneous dialog, can still require a DSA 'end of group' before they will handle an incoming message.

Note that most *Host Links* products and host applications are not in fact two-way-simultaneous in the sense that they send and receive simultaneously at the application level. Two-way-simultaneous means in practice that the application or communications product can break the 'send, receive, send, receive' pattern of interactive dialog. Thus an emulator or concentrator connected to a DMIV TP application using DACQ configured for two-way simultaneous dialog will be able to send a second transaction before it receives an answer to the first. In general this allows applications to not answer certain transaction types (data entry) for efficiency.

Host Links products by default always offer to establish either two-way simultaneous or two-way alternate sessions, and it is the mainframe application that chooses the mode.

Host Links Trace

If you experience any kind of problem when using a Host Links application, the application trace file and/or the line handler trace file will provide useful documentation of the problem.

Trace activation

The Host Links products automatically create sub-directories in the debug directory when debug is activated: at product level using the `-dbg` parameter, or at line level using the `-d_` or `-s_` parameters to the line module.

Windows server	`gspool -id gs1 -dbg -ps \\SERVER\LEXMARK -li dsa -da tptst -d_ on`
UNIX Linux	`gspool -id gs1 -dbg-pc lp -li dsa -da tptst -d_ on`

Most G&R products include a facility for setting product or line parameters dynamically. It is therefore generally possible to turn on debug or trace without modifying the command line or configuration of a production system.

Trace types

All Host Links products accept a parameter `-dbg`, which starts an application level trace of internal events. This is useful when investigating malfunctions or looking closely at product behaviour.

All Gline line handlers accept a parameter `-d_` to turn on a data trace. It records data and enclosure level being exchanged with the line handler. It is useful when documenting product malfunction e.g. an emulation error, because it records exactly what the host sends and what the G&R application replies. It can be used to simulate a customer situation, reproduce a problem and to verify that a correction fixes the documented problem.

All Gline line handlers accept a parameter `-s_` to turn on a session trace. It records the raw data being exchanged between the line module and the underlying transport layer (e.g. OSI Transport, or TCP socket), as well as internal events and protocol states. It is useful when investigating protocol failures such as unsuccessful connect attempts or abnormal disconnections.

Structure

The Host Links file structure includes a debug directory to collect the trace and debug files in one location where the permissions can be adjusted as required for security. By default only the Host Links administrator can access the directory. The debug directory is created by the initialization procedure and located (by default) in:

Windows server	`\gar\debug`
UNIX Linux	`/usr/gar/debug`

If the application is a client type of application, a debug sub-directory with the same name as the user (UNIX username or PC login name) is created and all debug files are located there. This includes the line level trace except in the special case where the client application connects via Ggate and the line level trace is written on the Ggate system using the Ggate DSA node name as a debug sub-directory.

If the application is a server type of application, then a sub-directory will be created using the DSA node name on behalf of which the server application is executing. If the server does not use DSA the default local session control name is still used if there is a `dsa.cfg` file. If there is no `dsa.cfg` file then the system's UNIX or Windows communications node name is used. You can find this name using the command `uname -n` on UNIX systems, or the Network section of the control panel on Windows systems. This covers situations where several instances of a server are executing on the same system and accepting incoming calls to different DSA node names, or where several Host Links systems using the same server product share a file system.

Tracing Ggate

When Glink, a Host Links client or a customer application based on GlAPI connects through Ggate to the application, the line handler trace is generated on the Ggate system, with the name and location shown in the table:

Windows server	\gar\debug\NODE\gga**NN-PPPP**.dbg
UNIX Linux	/usr/gar/debug/NODE/gga**NN-PPPP**.dbg

NODE is the local DSA node name used by the Ggate system.

The trace file name consists of the prefix `ggaNN-` followed by the IP-address of the client, suffixed by `.dbg` for a terminal session or `-dbp` for a printer session. The following is a trace file name for Ggate session sequence number 5 executing on Host Links system `GRDL` initiated from a Glink client on IP-address `jim.gar.no`:

```
gga05-jim.gar.no.dbg
```

This file, and possibly also a Glink debug file and a Glink communication trace file activated by the `/J` command line parameter will be needed by the support engineer investigating any problem.

To enable a line handler trace through Ggate the product's start-up command or configuration file would look like this:

```
-LI YYY:ZZZZ -S_ -D_
```

(YYY =line handler identification, i.e. DSA or DIWS)
(ZZZZ =IP-address of the system running Ggate)

Examples - G&R products

Examples of directory and file names in the debug structure are:

/usr/gar/debug/jim	**Debug directory for user 'jim'**	
qsm.dbg	Qsim emulator debug file	-dbg

qsm-gli.dbg	Qsim host line trace	-li dsa -s_
pth-glit.dbg	Pthru terminal line trace	-term -s_
pth-glih.dbg	Pthru -host line trace	-li dsa -s_
g32.dbg	G3270 emulator debug file	-dbg
g32-gli.dbg	G3270 host line trace	-s_
/usr/gar/debug/mike	**Debug directory for user 'mike'**	
v78.dbg	V78sim emulator debug file	-dbg
v78-gli.dbg	V78sim host line trace	-li dsa -s_
guf.dbg	GUFT client debug file	-dbg
guf-gli.dbg	GUFT client host line trace	-li dsa -s_
/usr/gar/debug/en01	**Debug directory for node 'en01'**	
guf.def	GUFT server debug file	-dbg
guf-gli.def	GUFT server host line trace	-li dsa -s_
gli-gli.dsa	DSA listener host line trace	-s_
gli-gli.diw	DIWS listener host line trace	-s_
gsp.def	Gspool (default -id) debug file	-dbg
gsp-gli.def	Gspool (default -id) host trace	-li dsa -s_
gga01-mike.gar.no.dbg	Ggate line trace, first Glink	-s_
gga02-mike.gar.no.dbg	Ggate line trace second Glink	-s_
/usr/gar/debug/en02	**Debug directory for node 'en02'**	
gsp.abc	Gspool (-id abc) debug file	-dbg
gsp-gli.abc	Gspool (-id abc) host trace	-li dsa -s_
gspc-gli.def	Gspool DPF8 command trace	-li tcp -s_

gspd-gli.def	Gspool DPS8 data trace	-li tcp -s_
gsp._00	Gspool started on demand debug	-dbg
gsp-gli._00	Gspool started on demand trace	-li dsa -s_

CPI-C and Gweb trace files

Gweb uses the CPI-C libraries so the Gweb debug structure is exactly the same as for CPI-C, except that Gweb inserts its own product identifier into the file name structure. CPI-C applications use the 'client' style of debug and create a debug directory with the UNIX username or PC login name used by the process that started them.

The application level debug (-dbg) and line trace (-s_ and -d_) are set in the cpic.cfg file. The line trace goes to the debug directory, with the name built up as follows:

```
<product_id><session_id>-<process_id>.<debug_type>
```

product_id	*Value*	*Comment*
	cpi	CPI-C API
	cp3	CPI-C 3270
	cp7	CPI-C 7800
	cpd	CPI-C DKU
	gw3	Gweb3270
	gw7	Gweb7800
	gwd	Gwebdku
session_id	(nn)	If multi-session application, 1-63
process_id	n (n n n...)	Varies by platform
debug_type	dgb	Application level debug
	gli	Line trace

Example:

\gar\debug\system	debug directory for user "system"	
cpi-16.dbg	CPI-C single session debug	-dbg
cpi-16.gli	CPI-C single session line trace	-li dsa -s_
cpi2-123.dbg	CPI-C session 2 application debug	-dbg
gw7-20172.gli	Gweb7800 host line trace	-li dsa -s_

Line handlers

The DSA/DIWS handler

You start this handler when you specify `-li dsa` or `-li diws` as the line handler for a *`Host Links`* product. If you connect via *`Ggate`* it creates a new line handler process or thread for you, and the new process/thread runs `gl_dsa` on your behalf.

The handler is used to connect to Bull front-ends or Bull systems directly:

- Directly to Datanet, MicroFEP or MainWay
- Through Datanet, MicroFEP or MainWay to any mainframe
- Directly to GCOS7 systems with ISL
- Directly to GCOS7 systems with FCP7
- Directly to the Open Systems personality of GCOS7 Diane systems
- Directly to GCOS8 systems with FCP8
- Directly to GCOS8 systems with GNSP
- Directly to GCOS6 (WAN or LAN)

The DSA/DIWS line handlers can (like all others) be used in communication between two Host Links applications; for example for test purposes or when running *`G&R/GUFT`* file transmission between 2 systems.

The DSA line handler in DSA protocol mode is the preferred way to connect to Bull systems. It implements DSA presentation over DSA session, interfacing to the transport stack. The main benefits of the DSA protocol compared to DIWS are:

- Access to front-ends/hosts that don't have a PID (Plug ISO-DSA). This includes many older releases of DNS as well as GCOS6 and GCOS7 software that is still in use at many sites

- Increased speed since the unneeded level of protocol translation is by-passed

- Simpler DNS and/or mainframe configuration, as it's likely that DSA session is already configured whereas DIWS is not

- The DIWS protocol mode can still be used if you have historic reasons to do so. It implements DSA presentation over OSI session, interfacing to the OSI transport stack according to the Bull DSA/ISO workstation specification. There are no known technical reasons to use DIWS in new installations

DSA/DIWS on Windows servers

Due to limitations in the multitasking capabilities of Windows servers the DSA/DIWS line handler has been implemented slightly differently than the UNIX/Linux counterpart. On UNIX/Linux, when you specify `-li dsa` or `-li diws` for a *`Host Links`* application, a new line handler process `gl_dsa` is launched. Windows servers cannot multitask multiple processes efficiently, and so need to handle multitasking by using threads within a single process. The Windows server version of *`Host Links`* uses a line module server, to deal with this. The line module server is a program with the same name as the listener products on UNIX/Linux, i.e. `nl_dsa`. When you specify `-li dsa` or `-li diws` for a *`Host Links`* product running on a Windows server, it contacts the appropriate line module server. The line module server creates a new thread to handle the session.

As a result of this, the `nl_dsa` line module server **must** be started before any *`Host Links`* product that uses the DSA/DIWS line handler. Like all the other *`Host Links`* servers on Windows, the line module server must be launched as a Windows service. Do this by adding the command line to the beginning of the Gservice configuration file:

```
nl_dsa.exe
```

Transport stacks

The DSA/DIWS handler interfaces to a transport stack. The transport stack traditionally used by Bull front-ends and mainframes was the OSI-transport stack, and the Host Links products needed a corresponding OSI-transport stack in order to connect. This is no longer the case. Bull supports the Internet standard RFC1006 protocol layer, allowing native DSA sessions to be established over TCP/IP networks.

The DSA line module includes both the OSI interface and the TCP/IP based RFC1006 interface. On some UNIX systems OSI interface files are requested at load time and if no OSI stack is installed, the loading of the line module may fail. For this reason we deliver a DSA line module that includes only the RFC1006 interface and this module is optionally installed during the Host Links installation process:

```
Your choices are:
1. DSA over OSI and RFC1006 (requires OSI stack)
2. DSA over RFC1006 only
Your choice is [2]?
```

Please choose 2 if you will be using only RFC1006.

RFC1006

For customers with a MainWay front-end and Open Network Processor, a GCOS8 GNSP processor or a GCOS7 Diane system, RFC1006 support is standard. RFC1006 support can also be installed in the FCP7 or FCP8 FDDI adapters. For customers with RFC1006, the G&R products can make genuine, fully functional DSA connections over a TCP/IP network. This removes the requirement of purchasing and installing OSI transport software on the Host Links platform. We supply the RFC1006 protocol stack as a part of the standard delivery of Host Links. If RFC1006 is available on the mainframe system or MainWay front-end, then no communications software other than the standard TCP/IP stack is needed on the Host Links platform. See the `-who gar` parameter to the `tp` (transport provider) directive in the `dsa.cfg` file described on page 80.

If you intend to accept incoming connections over RFC1006, you **must** start the `nl_dsa` listener, and it **must** be started from `root` because only `root` can reserve the RFC1006 reserved port 102.

OSI-transport

The Host Links product set does not include an OSI-transport stack as a part of the standard delivery. If one is needed it must be ordered. We do not offer an OSI-transport stack for Linux; RFC1006 is the only choice. For most UNIX platforms an OSI-transport stack is available from the UNIX supplier. The great majority of the OSI stacks for UNIX systems come with a programmatic interface to the OSI-transport layer following the TLI, XTI or ISODE API standards, all of which we support. For Windows servers we supply the Marben OSIAM stack as an extra option. For OSI connections over Ethernet nothing further is needed, the Marben stack supports all third party Ethernet cards with the Windows NDIS interface. For OSI connections over a WAN an X.25 card is needed. For Marben you will need an EICON X.25 board. See the ***Host Links installation and Configuration*** manuals for UNIX/Linux and Windows servers for details on installation and configuration of OSI stacks for various supported platforms.

Accepting connections

Products and applications receiving connections must configure a local node name using the `-ln` option, causing the line handler to accept incoming calls. The local mailbox name should then also be specified using the `-mn` option. Because a line handler can deal with only one incoming connection, a system accepting multiple simultaneous connections to a single node name **must** use a listener, described below. The `-ln` option then registers the application with the listener for the given node name as listening on the given mailbox. It is then the listener for the given node name which accepts connections and forwards them to the applications which have registered that they are listening to mailboxes on that node name.

The listener

The `nl_dsa` program listens for incoming connections on a given `SC` (Session Control name, also referred to as DSA node name), and dispatches incoming connections to waiting programs. This allows you to have many programs listening for incoming connections on the same `SC`, but for different mailboxes.

We recommend you always use the DSA protocol (default) to avoid unnecessary overhead and many known errors in the DIWS handlers on the mainframes and front-ends, but you can use the DIWS protocol if you must.

Please note that a listener can listen for either DSA or DIWS connections, but not both protocols at the same time. If you need to accept both DSA and DIWS connections, then you need to start up 2 instances of the `nl_dsa` program: one with the `-prot DIWS` parameter, and one with default (`-prot DSA`). Also note that if RFC1006 protocol is used, 2 listeners cannot use the same RFC1006 port number; you must start one of the listeners using a non-standard port number (i.e. different from 102) using the `-lp` parameter.

Listener parameter overview

Option	value	Explanation
`-prot`	DSA DIWS	Session protocol to be used. Default DSA
`-dbg`		Enable data and event tracing. Write information to the debug file `gli-gli.XXX` in the default debug directory
`-e_`		**Windows** server version only: enable Windows Event Log Tracing
`-icolim`	n 50 200	Number of simultaneous incoming sessions accepted by the **Windows** listener before it begins to reject them in order to avoid unpredictable errors due to Windows service resource limitations. The default is 50. You can increase the default to 200 by running the Host Links Windows services in the foreground. Use the 'foreground' parameter to Gservice (see the Gservice documentation in the Host Links for Windows installation manual).
`-icorej`	xxxx	When the **Windows** listener begins to reject incoming sessions because it has reached the limit `-icolim` it will by default use DSA reason code 0106 (mailbox saturated). You can change it to e.g. 0105 (mailbox inoperable).
`-ln`	SCID	Set if multiple SC records are defined and you want to listen for incoming sessions on other than the default SC. Also set if you want to run several listeners
`-lp`	Port number	Local port number (applicable to RFC1006). Default 102
`-rej`	Reason code	Overrides the DSA reason code used when rejecting an incoming session connect from the mainframe. Defaults to 0104 for unknown mailbox and 0106 for busy mailbox

Using the Windows listener

For Windows servers the `nl_dsa` line module server (listener) **must** be started before any *`Host Links`* product can use the DSA/DIWS line handler, see the section describing use of the DSA/DIWS handler on Windows servers on page 18. The same line module server also carries out the listen function, and so any product using the `-ln` option registers the mailbox for which it will accept connections with the line module server (listener) that is handling connections for the node name specified in the `-ln` parameter.

Using UNIX/Linux listeners

If you need more than one listening application on a UNIX/Linux system, either using the `-ln` option directly or implicitly through the `-co` or `-pco` options explained below, then the listener process, `nl_dsa`, has to be started. It can be started manually, but typically you should set it up such that the command is run each time the UNIX/Linux platform boots. This process has to be started before any other *`Host Links`* product that receives incoming connections. If the DSA listener is accepting incoming RFC1006 connections, it must run as `root`, because access to the RFC1006 port is restricted to `root`.

On systems running AIX or HP-UX, you can do that by starting it directly from `/etc/inittab`, by placing a line like this at the end of the file:

```
nl_dsa:2:once:/usr/gar/bin/nl_dsa
```

On other UNIX or Linux systems and you can create a shell script, e.g. `S99nl`, with the content below, and place it in the `/etc/rc2.d` directory:

```
/usr/gar/bin/nl_dsa
```

The `nl_dsa` program accepts a node name parameter, `-ln nodename`. This option needs only be specified if you have defined multiple `SC` records in your `dsa.cfg`, and want to listen for incoming connects on other than the default `SC`. This would be the case if you were accepting incoming calls to multiple node names and needed multiple listeners.

The listeners also accept the line session trace parameter `-s_`, which causes them to produce a line trace in the debug directory where they record the connections established through the listener.

Mailbox pools for incoming connects

Contrast this functionality with '*Starting applications on demand*' described below.

You can start multiple instances of products accepting connects to the same mailbox name on the same DSA node (`SCID`). Each instance should use a different mailbox extension. The instances of the product will be allowed to start execution, register the mailbox name and extensions for which they are accepting connects, and act as a mailbox pool for applications connecting to the mailbox name.

An incoming connect to a mailbox without an extension will be passed to a random instance of the product which is accepting connects to a mailbox with the correct name and that isn't currently busy handling a session. An incoming connect with a mailbox and extension will only be connected to a product which is accepting connects that match both.

Example:

```
gspool -id gs1 -pc "lp -dprt1" -li dsa -ln -mn printer1 -mx p1
gspool -id gs2 -pc "lp -dprt2" -li dsa -ln -mn printer1 -mx p2
gspool -id gs3 -pc "lp -dprt3" -li dsa -ln -mn printer1 -mx p3
```

The three instances would start, all listening on mailbox `printer1`. Connects to mailbox `printer1` would be accepted by any of the Gspool instances that was not already busy. Because the instance that is selected is random the three instances should deliver print to a printer or printer queue with the same print characteristics. If a certain application needs a specific one of the printers or print queues that is otherwise used in the pool, then that application can use the correct mailbox extension. Print output needing another type of printer altogether would have to connect to another mailbox name, serviced by one or more Gspool instances with a different printer or printer queue.

Starting applications on demand

You can start applications on demand when DSA or DIWS connections arrive. All the G&R server-type applications that accept incoming connections (e.g. Gspool, GUFT and Gmailer) can be started in this way, as well as user-written applications using GlAPI.

Start on demand is especially useful for sites that need to emulate many printers. Pre-starting one Gspool process per printer is an administrative problem for sites that have hundreds of them.

You can also accept multiple simultaneous incoming connects to GUFT servers with the same Host Links node name and mailbox name.

The `nl_dsa` listener handles the implementation; so this must be started. A configuration file controls the mapping from the DSA mailbox name (`-mn`) and, optionally, extension (`-mx`) to the command line for the application to be started. There is one file for each DSA node name (`SCID`) for which connections are being accepted. It is placed in the directory:

```
/usr/gar/servers/<scid>.gli
```

The file is either `config.dsa` or `config.diw` for DSA protocol and DIWS protocol respectively.

Example file `/usr/gar/servers/grdl.gli/config.dsa`:

```
* Three printer mailboxes
listen -mn printer1 -cmd gspool -pc "lp -dprt1"
listen -mn printer2 -cmd gspool -pc "lp -dprt2"
listen -mn printer3 -cmd gspool -pc "lp -dprt3"
* UFT server
listen -mn filetran -cmd guftsrv
* User written application
listen -mn userapp -lim 1 -ext -cmd userapp arg1 arg2
```

Explanation:

All blank lines and lines starting with # or * are ignored. The only directive currently handled is `listen`. It takes the following options:

`-mn`	Local mailbox name
`-mx`	Local mailbox extension
`-lim`	Used only if you want to enforce an upper limit on the number of simultaneous connections to this mailbox
`-ext`	Used to indicate that this is an external command, i.e. not a G&R product, but a user application written using GlAPI
`-cmd`	The command to be executed. It must be the last option on the line

The file above allows an unlimited number of simultaneous connections to mailboxes `printer1`, `printer2` and `printer3`. For each connection to one of the mailboxes a new instance of the corresponding Gspool is started, regardless of how many were already executing. Connections to mailbox `filtran` each start a new instance of *GUFTSRV*. Only one instance of `userapp` is allowed to start, new connections to `userapp` are refused until the first instance terminates.

If an application connects using both mailbox name and extension then there must be an entry that matches both. The entry will be used to start the application unless the limit (`-lim`) is reached. Two entries with the same mailbox name must have different extensions.

If an application connects using only the mailbox name then the first entry in the file with a mailbox name that matches will be used to start the application until the limit (`-lim`) is reached. If the limit is reached then the next entry with a matching mailbox name is used.

User written applications using this feature should be written like any other application using GlAPI, but be aware that:

- The session connection is passed on the standard input file descriptor, so this descriptor must not be changed.
- Additional information is passed using environment variables, so these must not be cleared.
- The application should terminate as soon as the session terminates. It should not itself go on to listen for further incoming connections. In other words, it should handle one session and then die. A new instance of the application will be started to handle the next incoming connect.

Issuing connections

When the handler issues a connect it asks by default for either a two-way simultaneous or alternate session, leaving it up to the acceptor to decide. TSS on GCOS8 would always choose alternate, but the DACQ gateway to DMIV TP or the CXI interface to TP8 can choose either, depending on Datanet and TP configuration. Two *Host Links* products communicating with each other would by default choose a two-way-simultaneous session.

Two-way-simultaneous sessions do not have a 'turn' concept, but the DSA session records indicate the enclosure level specified by the sender. Thus data blocks may be delivered with enclosure 'end of group' indicating 'turn'. GCOS8 DACQ applications using interactive dialog will generally indicate 'turn' at the end of their output even if they are configured as two-way simultaneous. GCOS6 command mode does not indicate 'turn'. Forcing two way alternate (`-CM TWAA`) is sometimes the only way to make some applications function correctly, and the *`Host Links`* products which have problems with 'turn' will do this by default.

The handler by default asks for an ASCII session when connecting to GCOS6 and GCOS8, and an EBCDIC session for GCOS7 and IBM hosts. When an ASCII product like Qsim uses an EBCDIC session to GCOS7 the code set is transliterated in the line handler. The transliteration can be modified to map ÆØÅ æøå 'correctly' for Scandinavian users (`-SX`). When *`G3270`*, *`G5250`*, the *`3270 CPI-C API`* or the *`5250 CPI-C API`* use an EBCDIC session they do their own transliteration.

Service messages

The line handler accepts service messages for connect `$*$CN`, disconnect `$*$DIS`, break `$*$BRK` (attention, and possibly purge undelivered data, depending on configuration of the mainframe, and/or application), the two attention signals `$*$A1` (attention type 1), `$*$A2` (attention type 2) and identification query `$*$ID` from the communications product using the handler. You can type them and transmit when using an emulator or concentrator. Unknown service messages (messages beginning with `$*$` but where the 2-3 following characters are not recognized) will cause an error message.

Interpretation of service messages within the message text can be disabled (`-DSD OFF` as an emulation parameter) so that they are transmitted as data to the mainframe. The emulators all have command keys for issuing service messages that may be used instead of typing `$*$` if interpretation is disabled.

User applications using one of our communications APIs from the *`GlAPI`* library must send service messages themselves when the *`Gline API`* is used, but they will be sent automatically by `CMALLC`, `CMDEAL` and `CMSERR` when one of our *`CPI-C API`*s is used.

The connect command

3 different `$*$CN` formats can be used:

Using a predefined CONAME from `dsa.cfg` to connect:

```
$*$CN coname
```

If the name given is not a predefined CONAME, it is assumed to be the application name and the connect will be carried out using the parameters set for the DSA session (i.e. explicitly by parameter setting from command line and/or the configuration file)

Using the traditional G&R positional `$*$CN` format:

```
GCOS8 connect to 'appl' (application)
$*$CN appl,node[,str,password,person,project,billing]

GCOS7 connect to 'appl' (application)
$*$CN userid,project,billing,appl,password,node[,,str]
```

Using the Terminal-Manager keyword connect format:

```
$*$CN -scid NODE -dmb APPL -ext . . .
```

The following key words can be used:

```
-dmb, -ext, -scid, -lmb, -str, -usr, -ba, -pj, -pw, -mdmp
```

A GCOS8 connect normally requires only application and node, but you may need to specify a GRTS ID or LID depending on Datanet configuration. This is supplied as `-str`. You will need `password`, `person`, `project` if the target site has the 'Secure Workstation Environment', or is using the IDCHECK package developed by Bull Norway for securing networks.

A GCOS7 connect normally requires you to set all of the documented parameters. The string (`-str`) `!NEW` may be used, and will force GCOS7 to start the dialog by presenting the top-level menu. Normally it reconnects at the level where you disconnected in the previous host session. Only a normal controlled disconnect (`BYE`) will get the top-level menu on reconnection.

Note that you can set the default values for the connect commands with parameters in the configuration file, but that if no default is configured then the last value used for this particular field becomes the default. So if you make an error in one of the parameters you need not specify the others in your next connect.

If you set all connect parameters in the configuration file you can connect from most of the *Host Links* products using a single key connect command, or by transmitting `$*$CN`.

Connecting to CXI applications

You must use `-HM CXI` if you log in to GCOS8 via CXI. When you log in the endpoint (at least for TP8) is:

```
node/mailbox/mailbox extension
```

When you define a `LID` in TP8 one of the parameters is mailbox extension, usually set to be equal to the `LID`. In the connect (`$*$CN`) the first parameter (mailbox name) can be specified as follows:

```
$*$CN XXXXXX YYYY,node
```

`XXXXX`	the TP8 mailbox
`YYYY`	the extension for your LID

When the remote mailbox string contains 8 characters you must follow the mailbox string with the mailbox extension string without a space separator.

Alternatively the remote mailbox is configurable as default application (`-DA`) and in the same way the extension is configurable as default extension (`-DX`).

The disconnect command

All systems accept:

```
$*$DIS
```

force disconnection (normally done by application).

The break commands

```
$*$BRK
```

sends attention type 1, against turn if necessary. The host, and/or application session may be configured to purge undelivered data when a break is sent.

The default configuration for break is:

GCOS8 Attention and purge
GCOS7 Attention only
DNS Attention only

For DSA300 applications (CXI applications such as TP8) an option 'data attention' may be negotiated. For these applications a data attention is used instead of attention when transmitting a break.

```
$*$A1
```

sends attention type 1 as for break, but never purges undelivered data.

```
$*$A2
```

sends attention type 2, no purge.

The identification inquiry

```
$*$ID
```

returns the local DSA session control name, the local mailbox name and the terminal type used e.g.

```
$$DIWS: Your ID is: SC:EN3D MB:D24701   TM:DKU7107
```

Connection errors

When a connection request from any of the *Host Links* applications fails to reach the host application, the handler will return an error message that will help you find the problem. The following sections list the three different classes of errors you may experience, and suggest where you should continue your investigations.

t_open error

XTI/TLI *t_open* errors typically mean that the OSI and/or XTI are not installed at all, or are installed incorrectly.

Transport disconnect

'Transport disconnect received' errors after a connection attempt typically mean that you have a physical problem with the link or a network configuration problem. Look up the error code displayed in the XTI/OSI documentation. If you are using the Bull OSI-stack on an AIX platform use the `pmaderror` command to get the error code explained. Check with trace tools on the UNIX side if any packets leave the UNIX system. The problem may possibly be on the other side if transport does not manage to establish connection with the session layer there. In this case some of the OSI stacks will deliver the X25 or DSA reason code, in which case see the section entitled ***Error codes***.

Session refused

Session refused or *session aborted* errors after a connection attempt typically mean that you have Front-end or mainframe configuration errors, or a mismatch between the Host Links and Front-end or mainframe configuration.

Look up the error code displayed in the OSI/DSA network system messages and return codes manual, see the section entitled ***Error codes*** or use the `Gerror` utility, see section entitled ***DSA test utilities*** on page 109.

Parameters for DSA/DIWS handler

The following parameters apply to this handler (defaults in uppercase):

Parameter	Description
`-LI DIWS`	Start gl_dsa in DIWS protocol mode
`-LI DSA`	Start gl_dsa in DSA protocol mode
`-LI DIWS:GATEWAY`	Start gl_dsa in DIWS mode through Ggate running on GATEWAY
`-LI DSA:GATEWAY`	Start gl_dsa in DSA mode through Ggate running on GATEWAY
`-AM SNM`	Application mode GCOS6 SNM
`GEN`	Application mode general
`AST`	Application mode Datanet AST

Parameter	Description
-AW on/OFF	Delay initial turn from host (in the session accept) until real data is received on the session.
-BRK AT/ATPGLT	Can be used to override the defaults for purge handling on break. AT gives no purge, ATPGLT gives purge. See section entitled *DSA/DIWS service messages*.
-CM xxxx	Connection mode to be used when connection is initiated. Defaults to offer all modes.
TWSI	Two Way Simultaneous, Initiators turn
TWSA	Two Way Simultaneous, Acceptors turn
TWAI	Two Way Alternate, Initiators turn
TWAA	Two Way Alternate, Acceptors turn
-CO xxxxxx	Connection name. CONAME in dsa.cfg
-CP DSA300	Connection protocol DSA300
DSA200	Connection protocol DSA200
-CS ASCII/EBCDIC	Explicit setting of code set (ASCII or EBCDIC)
-CW nn/30	Connect wait time in seconds before failing
-D_ on/OFF	Trace data exchange with transport layer
-D? xxxxxx -PW -PSSW	Default password
-DA xxxxxxxx -DMB	Default application mailbox for connect
-DB xxxxxx -BA -BILL	Default billing
-DN xxxxxx -SCID	Default node to which you will connect
-DP xxxxxx -PJ -PROJ	Default project

Parameter	Description
-DU xxxxxx -USR -USERID	Default userid
-DX xxxx -EXT	Default application mailbox extension 4 characters.
-DEE	Delayed End to end ACK
-EE on/OFF	End to end ACK
-FA on/OFF	Do not allow change of application
-FM on/OFF	Do not allow change of mailbox
-FN on/OFF	Do not allow change of node
-HM DPS8	Host type GCOS8
CXI	Host type GCOS8 with CXI
DPS7	Host type GCOS7
DPS6	Host type GCOS6
ROUTER	Host type GCOS6 NTM
DN	Host type Datanet application
HostLink	A Host Links system (UNIX/Linux/Windows)
IBM	Host type IBM via Datanet SNA gateway
-IE ON/off	Insert ETX/CR at end of message
-II nn/15	Inactivity timer in minutes before automatic disconnect
-IOC n	Periodically (every n seconds) update IO counts for the session. Reported by Ggate to Gproxy.
-LM T32782	Log mode, for host type IBM only
-LN xxxx	Local node name used when accepting incoming connection (if other than default local SCID)
-LT xxxx	Local node name used for outgoing connections (if other than local SCID)

Parameter	Description
-MN xxxxxxxx -LMB	Local mailbox name
-MX xxxx	Local mailbox extension.
-NA xxx yyy zzz	Send statistics to Gproxy network administration stations (up to 4 IP addresses may be given)
-NB on/OFF	Enable broadcasting for any/all Gproxy stations
-PCO xxxxxx	Associated printer session connection name
-PCN on/OFF	Associated printer session auto-connect
-PTS ON/off	Associated printer session turn simulation
-PL nnn	Page/screen length of terminal in lines (max. 255)
-SX on/OFF	Scandinavian transliteration
-S_ on/OFF	Trace session events and transport data
-TI xxxx	Terminal identity sent to host
-TM xxxxxx -MD -MDMP	Terminal type to sent to host. The list of recognized types follows:
TTY33 TN300	Async, echo/character mode, treated as TTY
TTU8221 TTU8223	Async, echo/character mode, treated as TTY
VIP7801 VIP7802	Async, echo/character mode, treated as TTY
TXT7801 TXT7802	Async, but in text/forms mode
VIP7804 VIP7805	Sync, text/forms mode
VIP7804V VIP7805V	Sync 72-line text/forms mode
VIP7814	Sync, text /forms. Unlimited attributes
VIP8800	8-bit ASCII terminal. Forces -CS ASCII
VIP9800	8-bit PLW for GCOS7 Diane. -CS EBCDIC
HDS5 TWS2255	Async HDS3/5/7, echo/char mode, treated as TTY
HDS5T TXT2255	Async HDS3/5/7 in text/forms mode

Parameter	Description
HDS7 TWS2255V	7800 8-bit 72-line ASCII terminal -CS ASCII
VIP7700 VIP7760	Early sync text/forms terminals. Replaced by VIP7800 and DKU
DKU7001 DKU7002	Async, treated as TTY
DKU7102	Async, but with forms and attributes
DKU7007 DKU7107	Sync, text/forms
DKU7007D DKU7107D	Sync, text/forms, supporting local forms
DKU7211 DKU7211D	Colour model, without/with local forms
DKU9107	8-bit PLW DKU for GCOS7 Diane. -CS EBCDIC
DKU722X PRT722X	Printers for DKU.
A2 TTU8126	A2 same as PRT722X. TTU is an async printer
PRT9220	8-bit printer for GCOS7 Diane. -CS EBCDIC
IBM3270 IBM3270A	EBCDIC/ASCII 3270 models
IBM3278 IBM3278A	EBCDIC/ASCII 3270 models
IBM3287	Printer
DSAIP	DSA information processor. DSA type is1001
HXnnmm	nnmm are four hex digits
nn	list, 10 for information processor, 20 for terminal
mm	DSA code of processor or terminal
HX0000	Some CPI-C application/application connections
BROWSER	Terminal for TDS-Web and Web8
-UD xxxxxx -UR -STR -STRG	Default ‘GRTSID/LID/user string’ for GCOS8
-WI nnn	Page/screen width in characters (max. 255)

DSA and DIWS parameters in detail

-LI DSA and -LI DSA:GATEWAY

Also `-LI DIWS` and `-LI DIWS:GATEWAY`

These parameters start the DSA/DIWS line handler and select the protocol. If the colon and a gateway name are included, the handler is started on the Ggate gateway with the symbolic name or IP address `GATEWAY`. It also signals that the parameters following are for the line handler. The first group of line handler parameters **MUST** be preceded by a `-LI` parameter.

Parameters to *`Host Links`* applications are positional in the sense that they must be delivered to either the application or to the line handler. Parameter `-USER` can be used to switch back to parameters for the application and `-HOST` to switch back again to the line handler. They must be repeated if necessary before each group of parameters.

-AM SNM/GEN/AST

```
v78sim -li dsa -hm DPS6 -am gen
```

This parameter sets the communication parameters needed for connections to applications that do not follow the normal pattern.

When host mode (`-HM`) DPS6 has been specified the default is `SNM`. This sets communication mode to `TWAI` (half duplex) and communication protocol to `DSA300`. The default is `GEN` (generic) for other mainframes. You can use `-AM GEN` to override the default for DPS6. The handler will then let the DPS6 side select between full or half duplex communication.

This parameter may also be used to set `AST` mode, which is needed to connect successfully with the `AST` mailbox on Datanets.

-AW on/OFF

Macro driven connects or applications using GlAPI sometimes start before the session is fully established because some host applications give Host Links products the turn, and then take it back, causing the macro to fail. This parameter causes the line handler to wait until the first message is received from the host before telling the application that the session is established.

-BT 1/2/3

This parameter changes the default for the break signal. The default for DSA200 is 'Attention 1' (`-BT 1`), with 'Attention 2' (`-BT 2`) used in special circumstances. The default for DSA300 is 'data attention' (`-BT 3`). Data attention is normally negotiated when needed by the mainframe, and this parameter is not necessary.

-CM xxxx

```
qsim -li dsa -cm TWAI
```

This parameter sets a connection mode to be suggested to the other party when the connection is initiated. Four connection modes are supported:

`TWSI`	Two Way Simultaneous Initiators turn. The applications can send without having 'turn'. The application establishing the session starts.
`TWSA`	Two Way Simultaneous Acceptors turn. The applications can send without having 'turn'. The application accepting the session starts.
`TWAI`	Two Way Alternate Initiators turn. The applications must have 'turn' to send. The application establishing the session starts.
`TWAA`	Two Way Alternate Acceptors turn. The applications must have 'turn' to send. The application accepting the session starts.

By default the handler offers to do both TWS and TWA and leaves the initial choice of turn to the other end. When Gline receives an incoming connect it accepts whatever the other end suggests unless you set the `-CM` parameter.

-CO xxxxxx

```
qsim -li dsa -co b8dcacce
```

This parameter sets a symbolic connection name. The line handler parameters are set in `dsa.cfg` for the `CONAME`. The line handler processes the `-CO` parameter by **first** resetting **all** line handler parameters to their default value, canceling any parameters set before `-CO`, and then expanding the `CONAME`. Any parameters **after** the `-CO` override the parameters set by the expansion, if allowed in `dsa.cfg`. Those of you familiar with DNS will notice many similarities between a connection name and the DNS correspondent object.

-CP xxxxxx

```
qsim -li dsa -cp DSA300
```

This parameter forces the connection protocol to `DSA200` or `DSA300`. For host modes: `DPS7`, `DPS8` and `UNIX` the default connection protocol is chosen by the host (historically it was `DSA200`). For host mode `DPS6`, `CXI`, `IBM`, `ROUTER` and `DN` the default is `DSA300`.

-CS xxxxxx

```
qsim -li dsa -hm dps7 -cs ASCII
```

This parameter sets the character set for the session. The default value is `EBCDIC` for IBM, and for GCOS7 except when using an ASCII terminal mode. It is `ASCII` for all other connections.

-CW nn

```
qsim -li dsa -cw 10
```

This parameter defines the time in seconds that Gline waits for a connection to complete before interrupting it and declaring it to be unsuccessful (default 30 seconds).

-D_ on/OFF

Windows Server	`gspool -id gs1 -ps \\SERVER\LEXMARK -li dsa -da tptst -d_ on`
UNIX Linux	`gspool -id gs1 -pc lp -li dsa -da tptst -d_ on`

This parameter enables the line handler's data trace. Data sent and received to the client application is written to a file in the Host Links debug directory. If the `-PCO` parameter is included an additional trace file is written for the printer session. The trace file generated when this parameter is set, can be used as input to various debug tools used by G&R support in order to reproduce e.g. emulations problems. See section entitled ***Host Links trace***.

Default connection parameters

These parameters set defaults for connect parameters that would otherwise have to be given in the connect command.

-D? -PW, -PSSW xxxxxx

```
qsim -li dsa -d? garpassw
```

This parameter defines the default password for the mainframe.

-DA, -DMB xxxxxx

```
qsim -li dsa -da tss
```

This parameter defines the default application mailbox for the mainframe application. It may be up to 12 characters in length. Characters 9-12 become the extension and can be set separately with `-DX`.

-DB, -BA -BILL xxxxxx

```
qsim -li dsa -db expensive
```

This parameter defines the default billing for the mainframe.

-DN, -SCID xxxxxx

```
qsim -li dsa -dn b8dt
```

This parameter defines the default node for the mainframe. It must be an RSC record in the `dsa.cfg` configuration file.

-DP, -PJ -PROC xxxxxx

```
qsim -li dsa -dp unixcn
```

This parameter defines the default project for the mainframe.

-DU, -USR, -USERID xxxxxx

```
qsim -li dsa -du mmluser
```

This parameter defines the default userid for the mainframe.

-DX, -EXT xxxx

```
qsim -li dsa -hm cxi -dx gr01
```

This parameter defines the default mailbox extension for the mainframe. The mailbox extension is used when connecting to CXI (GCOS8) subsystems such as TP8 (CXI requires that `-HM CXI` be set), and will normally contain the LID (Logical ID). The mailbox extension can be up to 4 characters long.

Flow control parameters

You can use the following parameters to 'throttle' host output, and prevent accumulation of large amounts of unprocessed data on the client side, where it might be lost in case of a client failure.

-EE on/OFF (DSA200 to GCOS8 only)

```
qsim -li dsa -ee on
```

This enables end-to-end-ACK. The line handler attempts to negotiate EEACK with the DAC or DAC-Q gateway. If successful Gline acknowledges DSA records that request it by sending the required ACK as an OSI session expedited message. EEACK is designed to delay further mainframe output until the current block has been processed. It is often used by applications that send print reports to slow ROP printers connected to the front-end, to stop accumulation of unprocessed print blocks. Normally G&R products can process blocks faster than the line speed so this is never necessary.

-DEE on/OFF (DSA200 to GCOS8 only)

```
gspool -ts -li dsa -dee on
```

The line handler attempts to negotiate EEACK with the DAC or DAC-Q gateway, but it converts it to a 'turn' based mechanism on the client side. If successful Gline will convert any received DSA record that requests an ACK to be an EGI (End of Group) record, giving the 'turn' to the G&R application. When the application replies Gline will ACK the previous record. This will stop accumulation of unprocessed host output in the Gline buffers.

Making connect parameters fixed value

The following parameters stop the user from changing connection parameters that are set in the configuration file.

-FA on/OFF

```
qsim -li dsa -da tss -fa on
```

This restricts connections to the default application name (`-DA appl`). It must be entered after the `-DA` parameter.

-FM on/OFF

```
qsim -li dsa -mn jim -fm on
```

This locks the mailbox name (`-MN mbxname`). It must be entered after the `-MN` parameter. The local mailbox name is used for unique identification of the terminal. It can be used to generate the GRID or LID in the Datanet.

-FN on/OFF

```
qsim -li dsa -dn b8dt -fn on
```

This locks the configured default mainframe node (`-DN node`). It must be entered after the `-DN` parameter.

General DSA parameters

-HM DPS8/xxxx

```
v78sim -li dsa -hm dps6
```

This sets the target mainframe type. You select a set of internal parameters that control the connection and dialog with the mainframe. The default is DPS8.

mode	Explanation
`DPS8`	Host type GCOS8 with DAC/DACQ Gateway. This mode is used when communicating with TSS and DMIV/TP.
`CXI`	Host type GCOS8 with CXI. This mode is used when communicating with TP8 and other CXI applications.
`DPS7`	Host type GCOS7. Used when communicating with GCOS7.

mode	Explanation
DPS6	Used when communicating with GCOS6 systems through SNM.
DN	Host type Datanet. Used when connecting to DNS applications.
ROUTER	Host type GCOS6 NTM.
HostLink	Used when communicating with another Host Links system. Gmailer and GUFT can make such connections.
IBM	Used when communicating with IBM hosts through the Janus/OSF gateway in the Datanet.

-IE ON/off

```
v78sim -li dsa -ie OFF
```

This controls insertion of ETX/CR at the end of messages. If the terminal is in the TTY class (e.g. VIP7801) CR is appended, otherwise ETX is appended. The default is ON for data sent to DPS6 and DPS8 mainframe applications.

-Il nn/15

```
v78sim -li dsa -ii 30
```

This sets an inactivity timer, which defines in minutes how long the session can be inactive before being automatically disconnected. By default sessions are never disconnected. If -ii is used but nn is not set it defaults to 15 minutes. It can be used at the DSA node level (RSC directive) in the dsa.cfg file to set the timer for all sessions to a specific remote node.

-LM T32782

```
g3270 -li dsa -lm d4c32782
```

This defines a log mode, and is used for IBM mainframes only.

-LN XXXX

```
gspool -li dsa -ln
```

This enables accept of incoming connections. The XXXX is optional and specifies the local node name; otherwise the default node name is taken from dsa.cfg. XXXX must match an SC record in dsa.cfg.

-LT XXXX

```
gspool -li dsa -lt grdw
```

This defines a non-standard local node name for outgoing connections. XXXX must match an SC record in dsa.cfg.

-MN, -LMB xxxxxx

```
qsim -li dsa -mn mike
```

This sets the local mailbox name. It can be up to 12 characters long. Characters 9-12 become the extension and can be set separately with -MX. By default the handler assigns unique mailbox names based on the program's process id (pid) in the form Dnnnnn.

The mailbox name can be used as additional user identification for security. On GCOS8 systems characters 3 and 4 may be used to generate a GRID (see -GRID and -NGRID parameters in the DNS configuration), or the first three (BCD TDS) or four (ASCII TP) characters may be used to generate the LID (see -LID and -NLID parameters in the DNS configuration). For IBM (OSF/Janus) connections it can be used to select a specific LU or LU pool.

-MX xxxx

```
qsim -li dsa -mn arild -mx gr01
qsim -li dsa -mn "arild   gr01"
```

This defines a local mailbox extension. The mailbox extension is a 4-character string that is appended to the 8 first characters of the mailbox name. The mailbox extension can be included in the local mailbox name (-MN) as shown, but quotes are necessary if spaces are needed to make the mailbox name 8 characters long.

-NA wwwwww xxxxxx yyyyyy zzzzzz

```
qsim -li dsa -na 192.150.211.11
```

This directs network administration information from the line handler to systems running *Gproxy*. Up to four IP addresses (space separated) can be included. See also the -NB parameter.

-NB on/OFF

```
gg_tcp -nb on (-li not necessary for servers)
```

This enables broadcasting of network administration information to systems running *Gproxy*. Broadcasts allow multiple, but unspecified, systems running *Gproxy* to monitor G&R products. This is especially important for servers such as *Ggate*, *GwebS* and *GlinkJS* for which *Gproxy* can load-balance and license-share.

-PCO xxxxxx

```
qsim -li dsa -co tp8_app -pco printsess
```

This associates a CONAME for a printer with the terminal session. The Bull mainframes often send print over an independent print session, which has to be associated with a terminal session so that the print can reach the terminal's printer. The printer's CONAME must be defined in `dsa.cfg` and must expand into the set of parameters needed to establish the printer session.

Glink, the *Host Links* emulators and *Pthru* can all accept print output directed by the GCOS mainframe to an independent print session. For example, print directed to a GCOS7 Twriter queue or TDS/TP8 LID defined as a ROP printer can be printed on a printer attached to a PC with *Glink* or to the configured print path of *V78sim* or *Qsim*.

-PCN on/OFF

This causes the associated printer (`-PCO`) to initiate the connection. The default is to wait for the mainframe to connect to the printer.

-PTS ON/off

This can be turned OFF if the host does not want to get the 'turn' back each time it sends 'turn' to the printer. Leave it to the default value in most cases.

-PL nnn

```
gspool -li dsa -pl 48
```

This specifies the screen or page length of the terminal in lines (max. 255). The default value for a screen is `24`, and for a printer is `0`. It is required by some mainframe applications that need to know the length of the screen (24 or 72). Some mainframe print spooling systems have to know the page length of your printer. You must then add it to the parameters associated with CONAME (`-PCO coname`) if you use an attached printer, or to the parameters for Gspool.

-SX on/OFF

```
qsim -li dsa -sx on
```

This selects Scandinavian transliteration tables for conversion between EBCDIC and ASCII. The -SX parameter transliterates EBCDIC #@$ to ASCII [\]. It is seldom used, even in Scandinavia.

-S_ on/OFF

Windows Server	`gspool -id gs1 -ps \\SERVER\LEXMARK -li dsa -da tptst -s_ on`
UNIX Linux	`gspool -id gs1 -pc lp -li dsa -da tptst -s_ on`

This enables the session trace in the line handler. It documents session, transport and internal events along with inbound and outbound data for the transport interface. If the `-PCO` parameter is used then an additional trace file will be written for the printer session. See section entitled ***Host Links trace***.

-TI xxxx

```
qsim -li dsa -ti T1234
```

This defines the terminal id, used by the line handler when connecting to the remote system. It can be used by the remote system to generate a LID. The default value is `T000`.

-TM, -MD, -MDMP xxxxxx

```
pthru -li dsa -tm vip7804
qsim -li dsa -tm hx2033
```

This selects the terminal mode that will be delivered to the mainframe. It tells the application which terminal presentation protocol you will use. See section ***Parameters for DSA and DIWS handlers*** for possible values.

If you need to use some other mode, and know its hexadecimal identification (see the DNS V4 - *Terminal Management Manual*), then you can configure it using `-TM HXxxxx`, where `xxxx` is 2 bytes in hexadecimal.

-UD, -UR, -STR, -STRG xxxxxx

```
qsim -li dsa -hm dps8 -ur glid
qsim -li dsa -hm dps7 -ur !NEW
```

This defines the default user record for the mainframe to which you are connecting. It can be 2-32 characters long, and is normally used to deliver the GRTS id (GRID) for connection to DAC programs on GCOS8, or the Logical id (LID) when connecting to TDS or DMIV-TP by DACQ.

When communicating with GCOS7 hosts the string `!NEW` may be used, which will force GCOS7 to start the dialog by presenting the top level menu. Normally it would have tried to reconnect at the level where you aborted in the previous host session.

-WI nnn

```
gspool -li dsa -wi 255
```

This parameter specifies the screen or page width of the terminal in characters (max. 255). The default value for a screen is `80`, and for a printer is `132`. It is required by some host print spooling systems that have to know the page width used in your printer, and must then be added to the parameters associated with `CONAME` (`-PCO coname`) if you use an attached printer, or to the parameters for Gspool.

The TCP/IP handler

You start this handler (`gl_tcp`) when you specify `-li tcp` as the line handler for a *`Host Links`* product. When you connect from one G&R product (or GlAPI application) to another the TCP/IP handler uses a 'raw' mode; the application data is simply packed in TCP packets and transmitted with no higher-level protocol involved. Terminal sessions use the Telnet protocol to reach standard Telnet servers, TN3270/TN3270E to reach TN3270 servers, TN5250 to reach TN5250 servers and TNVIP to reach TNVIP servers. Terminal sessions use Rlogin protocol when user identification is to be passed between UNIX/Linux client and server systems.

Making connections using TCP

Accepting a connect

If you configure a local port number (`-lp`), products and applications can use the TCP handler to receive connections.

Note that you must give the local port parameter at startup in the command line or configuration file for it to take effect. You can start two products on the same or two different systems and then connect from either to the other to establish interactive communication between two terminals or applications.

Issuing a connect

TCP connections in 'raw mode' are always two-way simultaneous. Any concept of 'turn' to determine initiative must be implemented in the applications. The 'end of data' marker is set following criteria depending on application mode, and is mapped to 'end of group' for the application using the handler. It can be interpreted as 'turn'.

For TNVIP, TN5250 and TN3270 connections the concept of turn is included in the protocol, and mapped to 'end of group' for the applications using Gline.

Service messages

The TCP handler accepts service messages for connect (`$*$CN`), disconnect (`$*$DIS`) and break (`$*$BRK`) from the communications product using the handler. You can type them and transmit when using an emulator or concentrator.

User applications using one of our communications APIs from the *GlAPI* library must send the `$*$CN`, `$*$DIS` and `$*$BRK` messages themselves when the *Gline API* is used, but they will be sent automatically by `CMALLC`, `CMDEAL` and `CMSERR` when one of our *CPI-C API*s is used.

The connect command

The only information required is the IP address in symbolic or numeric format and the port number if it is not the standard Telnet port e.g.

```
$*$CN hostname:portnumber
```

Make connection to 'hostname'.

Telnet, TN3270, TN5250 and TNVIP servers are all normally configured to use the standard Telnet port, and the TN3270, TN5250 or TNVIP dialects are arrived at by Telnet negotiation.

Note that you can set the default values for the connect commands with parameters in the configuration file, but that if no default is configured then the last value used for this particular field becomes the default. So if you make an error in one of the parameters you need not specify the others in your next connect.

If you set all connect parameters in the configuration file you can connect from most of the *Host Links* products using a single key connect command, or by transmitting `$*$CN`.

The disconnect and break commands

All systems accept:

```
$*$DIS
```

force disconnection (normally done by application).

```
$*$BRK
```

interrupt application, against turn if necessary.

Parameters for the TCP/IP handler

The following parameters apply to this handler (defaults in uppercase):

Parameter	Description
-LI TCP	Start gl_tcp
-AM XXXXXX	Application mode
RAW	Default for other ports than Telnet
CRTERM	
LFTERM	
TELNET	Default for the Telnet port
TN3270	
TN3270E	
TN5250	
TNVIP	
RLOGIN	
-AP on/OFF	Select the printer LU name associated with the LU name used by the screen (-LU)
-DN hostname:port	Numeric/symbolic IP-address, optional port
-D_ on/OFF	Data trace between application and host

Parameter	Description
-HO hostname:port	Numeric/symbolic IP-address, optional port
-HW n	Enable 'delayed enclosure'. Number of sec/10 to wait for more downstream data before 'end of group enclosure' is posted
-LP local_port	Numeric/symbolic: Incoming connections
-LU luname	TN3270 LU name
-DEV devicename	TN5250 device name
-NU on/OFF	Enable Telnet CRNUL mode
-RES mailboxname	Resource name for TNVIP
-RP remote_port	Numeric/symbolic: Outgoing connections
-SU username	User name for Rlogin
-S_ on/OFF	Session trace, internal events and data
-TM terminal_type	Telnet terminal type negotiation

TCP parameters in detail

-LI TCP

This starts gl_tcp. It also signals that the parameters following are for the TCP line handler. The first group of line handler parameters **MUST** be preceded by an -LI parameter.

Parameters to *Host Links* applications are positional in the sense that they must be delivered to either the application or to the line handler. Parameter -USER can be used to switch back to parameters for the application and -HOST to switch back again to the line handler. They must be repeated if necessary before each group of parameters.

-AM XXXXXX

This is used to select a mode of operation. The following modes are available:

-AM mode	Description
RAW	In this mode all data is passed to the application with an end of data marker.
CRTERM	In this mode all data buffers are marked with a 'more data' marker until a Carriage Return (CR) is found. When CR is received the data passed to the application is marked with an end of data marker.
LFTERM	In this mode all data buffers are marked with a 'more data' marker until a Line Feed (LF) is found. When LF is received the data passed to the application is marked with an end of data marker.
TELNET	In this mode the Telnet protocol is used for connection, negotiation/setting of Telnet options and data traffic.
TN3270	In this mode the TN3270 protocol (RFC1576) is used for connection, negotiation/setting of Telnet options and data traffic for a 3270 screen-session. LU name selection and print are supported using the TN3270 extensions in RFC1646.
TN3270E	In this mode the TN3270 extensions from RFC1647 are used to improve handling of screen and print sessions.
TN5250	In this mode the TN5250 (RFC1205) protocol is used.
TNVIP	In this mode the TNVIP protocol is used (RFC1921). The gl_tcp handler acts as a TNVIP client and communicates with a TNVIP server.
RLOGIN	In this mode the RLOGIN protocol is used for connection.

Telnet

```
v78sim -li tcp -rp 9998 -am telnet
```

The default application mode is `TELNET` for outgoing connections, unless connecting to a port number other than the `TELNET` default (`23`). If another port is chosen the default mode is `RAW`, which is also the default for incoming connections. If Telnet is required on a non-standard port it must be set.

TN3270 and TN3270E

Gweb, G3270 and Gspool use these modes. In TN3270 mode Gweb, G3270 and Gspool can select LU name. Gspool supports TN3270 print as specified in RFC1646 and used in the SNA gateway from Open Connect Systems.

In TN3270E mode G3270 supports the ATTN and SYSREQ keys as well as selection of a specific LU name. Gspool can be configured as a generic printer or associated printer through the use of TCP line handler parameters `-LU` and `-AP`. Microsoft's MS SNA Server (Windows NT), Bull's SNA/20 (AIX 4.1) and MainWay are examples of gateways that support TN3270E (RFC1647).

TNVIP

```
qsim -li tcp -am tnvip -tm dku7211
v78sim -li tcp -am tnvip -tm vip7814
```

The Bull TNVIP server on AIX uses a non-standard port, normally 7323, and requires parameter `-rp 7323`. The MainWay TNVIP server normally uses the default (`23`), but may need the `-res` parameter to select a specific configuration. When using TNVIP you must select one of the allowed terminal types, see `-TM`.

We have implemented the Telnet and TNVIP client protocols in the interest of completing our connectivity offer. Please note however that RFC1006 is supported by all MainWay front-ends with an ONP (Open Network Processor), and if used when communicating with G&R products it will increase throughput as compared to using TNVIP. It will also give a real, fully functional DSA or DIWS session over the TCP/IP network, as compared to the limited terminal session offered by TNVIP.

-AP on/OFF

Windows Server	`gspool -id gs8 -ps \\SERVER\LEXMARK -li tcp -am tn3270e -ho some.ibm.system -lu luname -ap -tm ibm-3287-1`
UNIX Linux	`gspool -id gs8 -pc lp -li tcp -am tn3270e -ho some.ibm.system -lu luname -ap -tm ibm-3287-1`

This tells the TN3270E server that the `-LU` parameter is the `LU` name of a screen, and it must allocate the printer associated with the screen's `LU` name in the TN3270E configuration.

-DN ipaddress:port

```
v78sim -li tcp -dn 192.150.211.4:9998
```

This selects the server to which you connect. You can use its numeric IP-address or you can use its symbolic name. In the latter case it must be registered in your hosts file or with your name server. You can append a colon and a numeric port number or use a symbolic name. In the latter case this symbolic name must be registered in your services file.

-D_ on/OFF

Windows Server	`gspool -id gs1 -ps \\SERVER\LEXMARK -li tcp -am tnvip -d_ on -ho some.gcos.system`
UNIX Linux	`gspool -id gs1 -pc lp -li tcp -am tnvip -d_ on -ho some.gcos.system`

This enables the line handler's data trace. Any data sent/received to the client application is written to a file in the Host Links debug directory. The trace file generated can be used as input to various debug tools used by G&R support in order to reproduce e.g. emulations problems. See section ***Host Links trace***.

-HO ipaddress:port

```
v78sim -li tcp -ho 192.150.211.4:9998
```

This selects the server to which you connect. You can use its numeric IP-address or you can use its symbolic name. In the latter case it must be registered in your hosts file or with your name server. You can append a colon and a numeric port number or use a symbolic name. In the latter case this symbolic name must be registered in your services file.

-HW interval (tenths of seconds)

```
v78sim -li tcp -hw 5
```

This sets a timeout for signaling 'turn'. It is needed when, after output, the server does not pass a turn signal to a Host Links client that needs it; typically a client simulating an asynchronous terminal connected by Telnet (e.g. *Gweb* connected to a HVX system). In the example, the line handler will pass the turn signal (enclosure 3) to the application when 0,5 seconds have passed without any further output from the server.

-LP local port

```
myglineapp -li tcp -lp 9999
```

This parameter selects the local port for incoming connections, and enables accept of incoming connections. You can use the port number or a symbolic name. In the latter case this symbolic name must be registered in your services file. The default is to not accept incoming connections. It defaults to raw mode.

-LU LUname

This is used in TN3270 mode to ask for a specific LU name on the host side.

-NU on/OFF

```
v78sim -li tcp -nu on
```

This enables Telnet `CRNUL` mode. Normally Telnet in ASCII mode encodes a carriage return (CR) as a Telnet 'new line' by appending a line feed (LF). In this mode a NUL character is appended instead.

-RES resource name

This parameter is used in application mode `TNVIP`. It selects a non-default 'mailbox name', which acts as a special access point in the TNVIP server. It has a maximum length of 12 characters, and will be appended to the terminal type (given by the `-TM` parameter), separated by a delimiter. It is used to select a specific terminal configuration in the terminal manager.

-RP remote_port

```
v78sim -li tcp -rp 9998 -am telnet
```

This selects a remote port for outgoing connections. You can use the port number or a symbolic name. In the latter case this symbolic name must be registered in your services file.

The default port number for the Telnet application modes, `TELNET`, `TN3270` and `TNVIP` is the 'Telnet' port number, `23`. If you set another remote port the default application mode is `RAW`. If you want to use Telnet you must set it.

-SU username

This sets the user name to be used when logging in to the Rlogin server.

-S_ on/OFF

Windows Server	`gspool -id gs1 -ps \\SERVER\LEXMARK -li tcp -am tnvip -s_ on -ho some.gcos.system`
UNIX Linux	`gspool -id gs1 -pc lp -li tcp -am tnvip -s_ on -ho some.gcos.system`

This enables the line handler's session trace. It documents session, transport and internal events along with inbound and outbound data sent and received through the underlying socket interface.

-TM terminal_type

This selects terminal type for Telnet negotiation performed in application-mode `TN3270`, `TN5250` or `TNVIP`. It is very important to select a terminal type acceptable to the server (IBM TCP/IP front end, generic TN3270⇔SNA gateway or the TNVIP server in a Bull MainWay, AIX gateway, GNSP or Diane).

The 3270 terminal types listed in *RFC1340 Assigned Numbers* are

```
IBM-3278-2          IBM-3278-3
IBM-3278-4          IBM-3278-5
IBM-3279-2          IBM-3279-3
IBM-3278-2-E        IBM-3278-3-E
IBM-3278-4-E        IBM-3278-5-E
IBM-3279-2-E        IBM-3279-3-E
```

The default value in application mode `TN3270` is `IBM-3279-2-E`.

```
g3270 -li tcp -tm ibm-3279-2-E
```

The supported 5250 terminal types are (see *RFC1205)*

```
IBM-5555-C01        IBM-5555-B01
IBM-3477-FC         IBM-3477-FG
IBM-3180-2          IBM-3179-2
IBM-3196-A1         IBM-5292-2
IBM-5291-1          IBM-5251-11
IBM-3812-1          IBM-5553-B01
```

The default value in application mode `TN5250` is `IBM-3179-2`.

```
G5250 -li tcp -tm ibm-3179-2
```

For TNVIP the terminal types listed in *RFC-1921* are

```
VIP7700             VIP7804
VIP7760             VIP7804V
DKU7005             VIP7814
DKU7007D            HDS7
DKU7105             VIP8800
DKU7107D
DKU7211
DKU7211D
```

DKU terminal types ending in D imply local form support. Local forms are not supported in our UNIX/Linux based DKU emulations (they are supported in Glink).

```
qsim -li tcp -am tnvip -tm dku7211
```

The X.25 handler

You start the X.25 handler (`gl_x25`) when you specify `-li x25` as the line handler for a *`Host Links`* product. It is only available for the UNIX/Linux version of *`Host Links`*, and only on some of the platforms.

Making connections using X25

Accepting a connect

The X25 handler by default does not accept connections because the 'outgoing only' parameter (`-OO`) is by default `ON`. If it is set `OFF` products and applications receiving connections can use the handler.

Note that the local sub-address parameter, the local user data parameter or both must be given at startup in the command line or configuration file to take effect and direct incoming calls to the correct product. You can start two products on the same or two different systems and then connect from either to the other to establish interactive communication between two terminals or applications.

Issuing a connect

X25 connections are always two-way simultaneous. Any concept of 'turn' to determine who may send data must be implemented in the applications. The handler recognizes the X.25 'more data' flag, and packets with this flag are delivered as enclosure 'none'. It delivers the terminating packet without the flag with enclosure 'end of group', and this may be interpreted as 'turn'.

Service messages

The X25 handler accepts service messages for connect (`$*$CN`), disconnect (`$*$DIS`) and break (`$*$BRK`) from the communications product using the handler. You can type them and transmit when using an emulator or concentrator.

User applications using one of our communications APIs from the *GlAPI* library must send the $*$CN, $*$DIS and $*$BRK messages themselves when using the *Gline API*, but they will be sent automatically by CMALLC, CMDEAL and CMSERR when one of our *CPI-C API*s is used. The PAD service messages can only be used from the *Gline API*.

The connect command

```
$*$CN remoteaddress,userdata
```

The only information required is the X.25 number to be called in numeric format, and the user data if needed in hexadecimal. Remote address is given as a numeric X25 subscription. User data is three hexadecimal bytes.

Note that you can set the default values for the connect commands with parameters in the configuration file, but that if no default is configured then the last value used for this particular field becomes the default. So if you make an error in one of the parameters you need not specify the others in your next connect.

If you set all connect parameters in the configuration file you can connect from most of the *Host Links* products using a single key connect command, or by transmitting $*$CN.

The disconnect and break commands

All systems accept:

```
$*$DIS
```

force disconnection (normally done by application).

```
$*$BRK
```

interrupt application, against turn if necessary.

Parameters for the X.25 handler

The following parameters apply to this handler (defaults in uppercase):

Parameter	Description
-LI X25	Start gl_x25
-AC on/OFF	Accept reverse charges
-CR on/OFF	Insert CR at end of message
-D_ on/OFF	Write Gline data trace to Host Links debug
-DS xxxxxx/NONE	Default Remote address.
-DU hxhxhx/NONE	Default Remote User data
-EB on/OFF	Use EBCDIC on line
-LU hxhxhx/NONE	Local User data
-HW n	Enable 'delayed enclosure'. Number of sec/10 to wait for more downstream data before an 'end of group enclosure' is posted.
-OO ON/off	Outgoing call only
-PD on/OFF	X29 PAD support
-RC on/OFF	Facility code: Make call with reverse charges
-RF nnn/NONE	Facility code: Receive throughput class override
-RS nnn/NONE	Facility code: Receive packet length override
-RW nnn/NONE	Facility code: Receive window size override
-S_ on/off	Write Gline session trace to Host Links debug
-SC nn/NONE	Local Sub-address
-SP on/OFF	Send data with even parity added
-UG nnn/NONE	Facility code: Closed User Group
-XF nnn/NONE	Facility code: Transmit throughput class override

Parameter	Description
-XS nnn/NONE	Facility code: Transmit packet length override
-XW nnn/NONE	Facility code: Transmit window size override

The Asynchronous (TTY) handler

You start the TTY handler (`gl_tty`) when you specify `-li tty` as the line handler for a *`Host Links`* product. It is only available for the UNIX/Linux version of *`Host Links`*.

Making connections using TTY

Accepting a connect

The TTY handler accepts connects in the sense that it sees the status of the line change if the other side (e.g. a modem) takes up the correct signals, and reports this as an incoming connect to the *`Host Links`* product.

Issuing a connect

TTY connections are always two-way simultaneous. Any concept of 'turn' to determine who may send data must be implemented in the applications. By default, TTY works completely asynchronously i.e. all data delivered to TTY is transmitted immediately on the line, and all data received is delivered immediately to the *`Host Links`* product with enclosure `none`. This applies even to single characters, so products running in 'echo' mode can use the TTY handler.

There are parameters for setting various control characters as the 'end of record' marker. These parameters implement 'block mode' operation. Incoming data is scanned for the end of record marker, and when found the accumulated data is delivered as an 'end of group' enclosure, which may be interpreted as 'turn'.

When sending data there are parameters to map enclosure level to control characters used as 'end of record' markers. This functionality is used to deliver the concept of enclosure and turn to our *`Glink`* emulator running on a PC when connected by an asynchronous line. This is necessary for some 'synchronous' emulation modes, particularly Questar DKU emulation.

Service messages

The TTY handler accepts service messages for connect (`$*$CN`), disconnect (`$*$DIS`) and break (`$*$BRK`) from the communications product using the handler.

User applications using one of our communications APIs from the *GlAPI* library must send the `$*$CN`, `$*$DIS` and `$*$BRK` messages themselves when the *Gline API* is used, but they will be sent automatically by `CMALLC`, `CMDEAL` and `CMSERR` when one of our *CPI-C API*s is used.

The connect command

There is no information needed when connecting to an asynchronous line. The TTY handler does not support dial-up modems.

`$*$CN`

Connect to the line. This will cause the correct signals to be given on the physical interface and the other side will recognize the line as active.

You can connect from most of the *Host Links* products using a single key connect command, or by transmitting `$*$CN`.

The disconnect and break commands

All systems accept:

`$*$DIS`

force disconnection (normally done by application).

`$*$BRK`

interrupt the application, against turn if necessary.

Parameters for the TTY handler

The following parameters apply to this handler (defaults in uppercase):

Parameter	Description
-LI TTY	Start gl_tty.
-BM on/OFF	Block mode. ETX/EOT used as end of record mark.
-CR on/OFF	CR mode. CR used as end of record mark.
-D_ on/OFF	Write Gline data trace to Host Links debug directory.
-DN XXXXX	Device name. Name of async port.
-EB on/OFF	Add ETB to send data when enclosure is e_message
-EX on/OFF	Add ETX to send data when enclosure > e_message
-HW n	Enable 'delayed enclosure'. Number of sec/10 to wait for more downstream data before an 'end of group enclosure' is posted.
-LF on/OFF	LF mode. LF used as end of record mark.
-NX on/OFF	Disable IXON/IOFF.
-S_ on/OFF	Write Gline session trace to Host Links debug
-ST XXXXX	stty options

DSA Configuration

Step-By-Step Guide

Configuration check list

Before you start to configure the mainframe and Host Links system you must obtain the configuration parameters you will need. If you already have a functional network, you do not need to change the configuration of the mainframe. Obtain the parameters from the network administrator.

The mainframe system

You will need:

- The DSA node name (SCID, session control ID) of the system

If DSA session is to be used for DSA200 connections:

- The DSA area address of the host (the TSN, e.g. 01:24)

- The network address of your mainframe system:

  If connecting over a TCP/IP network (RFC1006)

  - The IP address of the RFC1006 port on the FEP or mainframe

  If connecting over a LAN network:

  To a front-end:

  - The MAC address of the LAN adapter of the FEP

  Direct to the mainframe (ISL, FCP, GNSP, Diane):

  - The MAC address of the mainframe adapter

  If connecting over a private- or public X25 link:

  - The X25 address of the FEP to be used

The Host Links system

- The DSA node name (SCID, session control ID) for the Host Links system

  If DSA session is to be used for DSA200 connections:
  - The local DSA area address (the TSN, e.g. 05:36)

- The network address of the Host Links system:

  If connecting over a TCP/IP network (RFC1006)
  - The IP address

  If connecting over a LAN network
  - Ethernet MAC address

  If connecting over a WAN network
  - X25 address

Configuring the mainframe

GCOS8 configuration

There are no communication configuration files on GCOS8 unless an FCP8 controller is used. In that case you need to:

Set up the `osi.cfg`, `routes.wdf` and `loadscen` files as described in the section entitled ***Sample configuration*** (node B8DF) on page 74.

Configure the Host Links node with `nrsc` and `nrts` directives in the `osi-cfg`, for example:

```
nrsc name=grdl nrts=grdlts proto=dsa
nrts name=grdlts type=diws ivmo=fddi macaddr=0000C0E9A8E6
```

GCOS7 configuration

The network generation source file must be modified to include the description of the Host Links node. This is normally done with the `SYS` or `SYSTEM` macro directive, or with an `RSYS/RSC` pair of directives.

This can be done as follows for a DSA Host Links system:

```
RSYS NAME = GRDL, PF = 'STID/ISO/SID3', RSC = GRDL;
RSC NAME = GRDL, ADDR = DSA, SCID = 54:060, SVR=NCC1;
```

Or as follows for a DIWS Host Links system:

```
RSYS NAME = GRIL, PF = 'STID/ISO/SID3', RSC = GRIL;
RSC NAME = GRIL, ADDR = DSA, SCID = 54:060, SVR=NCC1,PRTC=SID;
```

Front-end configuration

If the connections go through a FEP, the FEP must have both the mainframe and Host Links systems configured. If you have an operational network the host is probably already configured (look for an `SC` directive with the DSA name of the host).

A host links node can be defined as being of type `-NAT ALL`, `-NAT ISO` or `-NAT DSA` and the network type can be `LAN` or `X25`. The following is an example of a Host Links DSA node with LAN access. For more detailed examples please refer to the section entitled ***Sample*** **`dsa.cfg`** on page 74:

```
SC GRDL RMT  -ADDR 49:002 -NAT DSA -SR GRDL
SR GRDL ISO  -TS GRDL
TS GRDL DIWS -NR GRDL -CLASS 4 -CRRE 15 -FEN 13 -TPDU 1024
NR GRDL LAN1 -PL GRDL
PL GRDL CSM1 -CB ETH1 -ETHAD 0000C0E9A8E6
```

Note that the `GRDL DIWS` in the example above means that the transport is ISO-transport (rather than native DSA-transport), it doesn't imply that the session protocol is `DIWS`. The `-NAT DSA` directive sets the session protocol to DSA.

GCOS6 configuration

Access to and from GCOS6 can go via a FEP or directly. In both cases, the Host Links system must be defined in the network configuration

`>sid>clm_net`	(Mod 400 v 3.1)
`>wdd>startup>nw_config`	(Mod 400 v 4.0/HVS1.0 and later).

An example of a LAN connected Host Links system follows:

```
SRISO     -RNSAP_NAME NSGRDLI
RSCA      -NAME GRDL -TYPE D6SC
RTSEL     -RNSAP_NAME NSGRDLI -T_SEL -HEX 4002454C3344 -TU_SIZE
          1024 -CLASS 4 -WINDOW 1 -MIN_CRED 1 -NO_CHECKSUM
RNSAP     -NAME NSGRDLI -OWNER ISOT -NSAP NSEN3CI
ROUTE     -NAME RTGRDLI -PROTOCOL CLNP -SUBLAYER LACS -NULL
          -LOCAL_SAP LPTP30 -REMOTE_SAP GRDLSAP
LLAN      -L_USER LPTP30 -R_USER LN3DSAP -FRAME 1497
LLAN      -L_USER LPTP30 -R_USER GRDLSAP -FRAME 1497
```

and in `>sid>lanc-f`

```
REMSAP    GRDLSAP,,X'20'/X'0000C0E9A8E6'
```

Another Host Links system

If you want to communicate with a G&R application on another Host Links node (typically GUFT/GUFTSRV or Gmailer), you need to configure the remote Host Link node in `dsa.cfg`. Refer to the next section for details.

Configuring the Host Links system

Configuring the transport stack

DSA is a session level protocol that runs over a transport stack. Traditionally DSA ran over the OSI-transport stack and a LAN or WAN network. Today the vast majority of networks are built of commodity products using the TCP/IP communications stack. To facilitate use of existing OSI-based applications over TCP/IP networks a 'Request for comment' (RFC1006) was submitted to the Network Working Group that manages the introduction of Internet standards. The RFC is entitled 'ISO transport services on top of the TCP' and has become an Internet standard. DSA runs over RFC1006 and TCP/IP networks.

RFC1006 transport

MainWay front-ends with an ONP (Open Network Processor) have RFC1006 support in the standard product, allowing DSA sessions over TCP/IP into the MainWay. RFC1006 can also be installed in the FCP7 and FCP8 cards to support DSA connections direct to the hosts without passing through the front-end. The GNSP communications processor of the later GCOS8 systems supports RFC1006. The Later GCOS7 Diane systems also support RFC1006.

If you use the RFC1006 protocol rather than OSI-transport protocols, you don't need any further communications software or configuration of the transport stack. We provide our own RFC1006 transport stack, which is controlled completely from `dsa.cfg`. RFC1006 is an Internet standard protocol that encapsulates an OSI-transport (TP0) protocol in TCP/IP frames. It is assumed that TCP/IP is already installed on the Host Links platform.

OSI transport

If you do not have RFC1006 the DSA connections can be made in the traditional way using OSI-transport, which is a requirement when connecting via old-style Datanets. The UNIX supplier typically delivers the stack, but we deliver the OSI stack on Windows NT/2000. The OSI stack must be installed and configured before the G&R Host Links products can run.

Some OSI stacks require that the DSA node names (or TSAPs) and or the network addresses of the local and remote DSA systems are configured for the stack. Other stacks do not need any of these parameters. For a description of the various OSI stacks and how to configure them please refer to the ***Host Links Installation and configuration*** manuals for UNIX and Windows.

Configuring Host Links

It is assumed here that the Host Links software, and OSI-transport, if used, is already installed on the target platform. For a detailed description of the installation process, see the ***Host Links Installation and Configuration*** manuals for UNIX and Windows.

The dsa.cfg file

You must configure the local Host Links system as well as the remote host systems you will access. For the local system you need an `SC` directive that contains the DSA session control name and the DSA address if DSA200 is to be used. The following assumes session control ID `grdl` and DSA address `54:60`.

```
sc grdl -addr 54:60
```

For each remote system to which you will connect you need a pair of `rsc`/`ts` directives. The following assumes a host named `b7dl` with a LAN MAC address of 080038177777 :

```
rsc b7dl -addr 1:4 -ts b7dl_lan
ts  b7dl_lan -class 4 -ns 080038177777 -tp grdl_lan
```

Finally for each remote system you need a description of the local entity (transport provider) that is used to reach that system. The provider is pointed to by the `-tp` parameter in the `ts` directive. The `tp` directive should have a `-who` parameter to identify the provider specifically, and has also an `-attach` parameter to define how the transport service is reached, normally it is the network address.

```
tp grdl_lan -who Bull -attach 02608c200441
```

The `dsa.cfg` file may also contain security filters and detailed connection parameters used when making defined host connections (`CONAME`). For a detailed description please refer to the section entitled ***The dsa.cfg configuration file*** on page 74.

Remember to compile the configuration file using the `glcc` compiler!

The listener

The listener is used for distribution of incoming connections from the remote host(s) to the Host Links applications. The listener must be used if several Host Links applications will receive incoming connects using the same local DSA session control name (TSAP).

The listener should be started automatically whenever the Host Links system is restarted. If the receiving applications are started before the connections arrive then no listener configuration is necessary. Each receiving application registers the mailbox name for which it will accept connects with the listener when the application starts.

If the session listener is to start the Host Links application on demand (when the DSA connection arrives), then you configure it in the listener's file:

```
/usr/gar/servers/NODE.gli/config.xxx
```

`NODE`	DSA session control name of the local system
`xxx`	protocol type (`dsa` or `diw`)

A sample directive is:

```
listen -mn filetran -cmd guftsrv
```

This would start up the GUFT server application whenever a request for the mailbox name `filetran` was received. For details see the section entitled ***The DSA and DIWS line handlers***, sub-section ***Accepting connections*** on page 21.

Application configuration files

Most of the host links applications have their own configuration files. These are located under the `/usr/gar/config` catalog structure and are used to minimize command line parameters. You do not need to set any of these until the connections are tested out successfully. These files are described in the application manuals.

Establish the physical connections

X25 connections

For a public X25 connection simply connect the appropriate cables to the X25 DCEs. Most DCEs have LEDs that indicate the state of the link. In most cases the link will be opened when the Host Links platform is started (i.e. before any connections take place).

For a private X25 connection, you can use a crossed cable (sometimes referred to as a W8 cable) where the output signals from the Host Links side are used to provide input signals for the host side. In this case one of the machines (and only one) must provide clock signals. Otherwise you need a pair of modem eliminator units (often referred to as MMEs).

Ethernet LAN connections

Connect the LAN adapters to the appropriate LAN segments, if necessary using media converters, tranceivers etc. etc.

FDDI LAN connection

Connect the FDDI adapter to the FDDI concentrator (either directly to a MainWay LAN Extender or some other external concentrator).

Test the connections

Make sure the configuration changes described above have been activated. The DSA configuration file on the Host Links system must be compiled. The system may need to be rebooted.

An external communications line analyzer tool (Sniffer or equivalent) is invaluable if you meet serious networking problems. However there are software tools that may help you analyze most problems.

Mainframe/FEP tools.

The availability of useful tools will vary with the system type and software version. Please check the mainframe documentation for information.

Most DSA systems have a Network Operator Interface (NOI) program. This is a very useful tool when testing out connection problems.

Host Links tools

If you are using OSI-transport there are usually internal debug and trace tools supplied with the communication stack. Check the documentation from the stack supplier, and see the platform-specific hints given in the ***Host Links installation and configuration*** manuals for UNIX and Windows.

There are several Host Links utilities and debug tracing routines you can use. You don't need to set up any Host Links application to do the initial tests, simply try

```
gping xxxx
```

or

```
gtrace xxxx
```

Both these utilities will attempt to connect to the DSA node with the name `xxxx` (must be defined in dsa.cfg). The Gtrace utility will trace the communication calls on the screen as the call progresses.

When starting any Host Links communication application, you can activate a session trace routine by supplying the `-s_` parameter. For example:

```
qsim ..... -li dsa ...... -s_
```

This will generate a session trace file in the Host Links debug structure. See the *Host Links Trace* section for more detailed information.

The dsa.cfg configuration file

Overview

One or more local session control entities (SC) must be defined. These are looked up using the value of the -ln line module option and contain the DSA session control name and number of the local UNIX/Linux or Windows server system, to be used by applications accepting incoming connects. The SC is also used to identify the calling Host Links node on outgoing connections.

Remote session control entities (RSC) must be defined. These are looked up using the value of the -dn or -scid line module options, the corresponding node name field in *Glink* or $*$CN commands in *Host Links* emulator products. They contain the DSA session control name and number of remote systems needed by applications making connects. They also define the transport station where the remote session control entity is to be found. If there are several paths to the remote session control entity several transport stations can be defined for backup or for load balancing.

Transport stations (TS) may be defined. They describe how to open transport connections to reach remote session control entities (DSA nodes).

Transport providers (TP) may be defined. They describe which transport stack to use on the local UNIX/Linux or Windows server system.

Symbolic 'connection names' (CONAME) may be defined. Clients then connect using these symbolic names instead of directly specifying a long list of connection parameters. This feature is similar to the 'correspondent object' in the Bull front-end configuration. Security filters (-FILTER) can be used to restrict access to these connection names.

A connection name may be defined so it picks up different connection parameters depending on which client is connecting to it, again by the use of security filters (PARAMS with -FILTER).

Connection names may also include access to pools of connection parameter lists, typically used to aid in generation of unique TP LIDs or GRTS IDs. These pools are also subject to security filtering (POOLS with -FILTER).

Access to undefined connection names may be configured to be either an error or to be ignored (COSTRICT).

Arbitrary options or groups of options may be restricted to clients passing a security filter (-FILTER).

The security filters referred to above are lists of IP addresses, IP network numbers, names of local hosts or names of applications. Each element in the list may either be permitted or denied access.

There is no limit on the number of objects that may be defined.

There is no limit on the size of any of these objects.

You configure these features by editing the dsa.cfg text file on the server. The file path is:

Windows Server	\gar\config\dsa.cfg
UNIX/Linux	/usr/gar/config/dsa.cfg

The formats of the directives in this file are described below. After a change has been made the glcc configuration compiler program must be run. This validates that the file is correct and copies/merges it into the file dsa.dat. It's the dsa.dat file that is actually used by the *Host Links* software at runtime. This file should **not** be edited directly. Always modify the dsa.cfg file and run glcc.

Note that editing the dsa.cfg file and running glcc while sessions are active is explicitly supported. The software has been carefully written to allow dynamic changes. You're even allowed to make changes in pool definitions where some entries are currently in use.

The Windows version of glcc has a -s option (silent mode). This option suppresses all message box displays, allowing for batch file installations.

Configuration file directives

In the syntax used below UPPERCASE is used for the fixed keywords and lowercase for the variable parameters. Brackets [...] are used around optional keywords. The vertical line | is used to separate alternatives. Symbolic names you define may be from 1 to 12 characters long and if they contain spaces the name must be enclosed in quotes. The order of the directives in the file is irrelevant except for the `PARAMS` and `POOL` cases mentioned below. Case is insignificant throughout the file. Space and tab characters can be placed anywhere between words and are ignored. Blank lines or lines starting with '*' or '#' are ignored. Comments can be put at the end of lines. The comment must be started with a # or * character.

SC - Session control

```
SC name [-ADDR x:x] [-SCID xxxx] [-DFLT] [-UNAME xxxx]
```

Defines a local DSA session control entity.

The name need not follow DSA name rules; it's only used as a lookup key for whatever is specified in the `gl_dsa -ln` option. In practice it will normally be the DSA session control entity name (node name). Normally you need only one of these directives.

`-ADDR` defaults to 1:1 and is optional. It's the DSA200 address (AREA:TSN) and need only be included if you will be using GL_DSA to make DSA200 connections. DSA200 is typically used for DPS7 native DSA access and for some DPS8 subsystems. The values are given in decimal.

`-SCID` defaults to be the same as the name when the name is 4 characters or less in length, otherwise `-SCID` is required. `-SCID` may in any case be specified to override the default. The SCID is used as the calling DSA session control name on outgoing connections.

`-DFLT` means this is the `SC` to use if a `gl_dsa -ln` option with no node name is used. Otherwise the first `SC` defined automatically becomes the default.

-UNAME is optional. It's the system's UNIX/Linux or Windows server communications node name. **NB!** it is not the DSA node name. This name can be displayed using the command uname -n on UNIX/Linux systems, and displayed in the Network section of the control panel on Windows server systems. When looking for an SC the software first looks for one with a matching UNAME, but if none is found it looks for one without a UNAME. It's only of use in setups where multiple UNIX/Linux and/or Windows server systems share a single dsa.cfg file on a file server.

```
sc is2c
sc is2c -addr 54:97
sc testsc -scid is2c
```

RSC - Remote Session Control

```
RSC name [-ADDR x:x] [-SCID xxxx] [-DFLT] [-II nn]
[-TS tstation [tstation] [tstation]]
```

Defines a remote session control entity (DSA node).

The name need not follow DSA name rules, it's only used as a lookup key for whatever is specified in the gl_dsa -dn parameter, in the node field in *Glink* or *Host Links* emulators $*$CN commands. In practice it will normally be the DSA session control entity name (node name). You need one of these directives for each remote node to which you connect or from which you will receive connects.

-ADDR defaults to 1:1 and is optional. This is the DSA200 address and need only be included if you will be using GL_DSA to make DSA200 connections to this node. DSA200 is typically used for DPS7 native DSA access and for some DPS8 subsystems. The values are given in decimal.

-SCID defaults to be the same as the name when the name is 4 characters or less in length, otherwise -SCID is required. -SCID may in any case be specified to override the default. This is the DSA session control name of the remote system.

-II nn sets an inactivity timer in minutes. Should the session be inactive for more than the time set here then the session is automatically disconnected. It defaults to 15 minutes if -ii is specified but nn is not given.

-DFLT means this is the RSC to use if no line handler -dn parameter, $*$CN nodename or Glink Host node is used when making a connection.

`-TS` points to a `TS` (Transport Station) directive somewhere else in the file. If you omit the option the default `TS` is used.

```
rsc is2b
rsc en3e -addr 54:62
rsc is2b -ii 15 -ts fep1_wan
```

To set up backup routes you just specify multiple `-ts` routes:

```
rsc -ts route1 route2 route3
```

These are then tried in order until a connection succeeds or the end of the list is reached.

Alternatively to configure load balancing you use:

```
rsc -ts route1 route2 route3 -lb 60 20 20
```

In this example `route1` gets 60% of the load, `route2` 20% and `route3` 20%. If the first priority route has a connect failure the second priority is tried etc. until a connection succeeds or the last priority is reached. A total of 8 routes can be specified.

The software remembers when a transport disconnection or time-out happens while trying to connect, and the node is suspended to avoid repeated connection attempts using a route that is down. The suspension time may be configured with the `-si` parameter to the `ts` directive.

If all routes to a given remote session control ID are marked as down then a connection is tried using the first priority route. If the connection succeeds that route is marked as being up again.

TS - Transport Station

```
TS name  -CLASS x -NS xxxx [-TPDU x] [-X25CALL xxxx]
[-X25FAC xxxx] [-EXP] [-NOEXP] [-SI nn] [-DFLT] [-TP tprovider]
```

Describes how to open a transport connection to reach remote session control entities (DSA nodes). You need one for each remote transport station to which you will be connecting. A transport station is the transport service access point, which can reside in a FEP (Front End Processor) or in the mainframes that have a direct connect capability, such as GCOS6, GCOS7 with ISL, GCOS7 or GCOS8 with FCP cards (FDDI adapters), GCOS8 GNSP or GCOS7 Diane.

-CLASS is required and must be set to 0 for RFC1006 connections, 2 for WAN/X.25 connections and 4 for LAN connections.

-NS is also required and for some stacks is a symbolic pointer into the transport configuration, but in others is the real network address. It (or the real address it points to) must be set to the X.25 address of the transport station for WAN/X.25 connections, the MAC or FDDI address of the transport station for LAN connections or the IP name or address for RFC1006 connections.

-TPDU defaults to whatever the local transport provider does and is optional. It's the maximum transport PDU buffer size. The actual size will be negotiated. Legal values are 128, 256, 512, 1024, 2048, 4096 and 8192. Some transport providers will ignore your TPDU size setting and always pick their own value. The maximum size for a LAN connection is 1024.

-X25CALL is optional and is a string of hexadecimal data that will be included as call data in the X.25 call request packet. This feature is only supported for the Bull XTI interface on AIX.

-X25FAC is optional and is a string of hexadecimal data that will be included as facilities in the X.25 call request packet. Only supported for the Bull XTI interface on AIX.

-EXP is optional. It forces the use of expedited data on transport stacks where we don't normally use it because of problems.

-NOEXP is optional. It suppresses the use of expedited data on transport stacks where we normally use it.

-SI is optional and specifies how many minutes to suspend after a connect failure; configuring 0 stops it from being suspended. The default is 15 minutes.

-DFLT mans that this is the default transport station. Otherwise the first TS automatically becomes the default.

-TP points to a TP directive somewhere else in the file. If you omit the option the default TP is used.

```
ts fep1_wan -class 2 -ns 123456
ts fep2_wan -class 2 -ns 888888
ts fepl_lan -class 4 -ns 080011223344
ts fepm_lan -class 4 -ns 080011223333 -tp lan
```

Deriving -NS for FDDI hosts

All Ethernet MAC addresses are in 'LLC' format. All FDDI addresses are in 'SMT' format. In SMT format you will see the address in the MSB (most significant bit) form. You will however normally, for instance in the case where you connect via an Ethernet port of a LAN Extender, see the LLC format only.

If you connect to an FDDI adapter directly or by an FDDI port of a LAN Extender, you must convert the MAC address to SMT. The way you do this is to read each octet backwards as shown below:

Example FCP8 Ethernet MAC address = 080038101054

```
08 = 0000 1000 binary, backwards: 0001 0000 = 10
00 =                                          00
38 = 0011 1000                    0001 1100 = 1C
10 = 0001 0000                    0000 1000 = 08
10 =                                        = 08
54 = 0101 0100                    0010 1010 = 2A
```

So the address to use here is 10001C08082A. A utility, Gmacfix, to convert Ethernet (LLC MAC) format addresses to SMT format is included in the Gline package. See the chapter on *DSA test utilities* on page 109.

TP - Transport Provider

```
TP name  [-WHO BJAMES | BULL | EICON | GAR | HP | MARBEN | SUN ]
[-ATTACH xxxx] [-DFLT] [-UNAME xxxx]
```

Defines a local transport provider. Although not strictly necessary on some platforms when there is only one stack installed we advise that it is configured with a minimum of the `-WHO` parameter to define the provider.

`-WHO` explicitly defines the provider.

`-ATTACH` is used to instruct the software how to attach to the transport service.

For the G&R/RFC1006 transport (all platforms):

Specify the value '`rfc`' in order to signal the use of RFC1006 over TCP transport. Please note that in order to be able to accept incoming connections when using RFC1006 you must start the DSA listener, even if only one program is accepting connections, and it must be started from root, since only root can access port 102, which is reserved for RFC1006.

For Marben on Windows servers:
Specify either `WAN` or `LAN`.

For SUN
Specify a local network layer address as configured in the local OSI stack.

For Bull:
Specify the X.25 or MAC address of the local OSI-transport interface you will be using. Alternatively give the value '`rfc`' in order to signal the use of RFC1006, and select the Bull-supplied RFC1006 (NetShare).

On AIX it is also legal to give a value of the form `pvc-xxxx` to signal the use of the X.25 PVC named xxxx.

`-DFLT` means that this is the default transport provider. Otherwise the first `TP` defined automatically becomes the default.

`-UNAME` has the same use here as in `SC`.

Examples of use on the Bull XTI interface and the Sun Solaris 2.x platforms:

```
tp wan -attach 123455
tp lan -attach 08001123355
```

Examples of use on SCO UNIX or System V release 4 on Intel with multiple OSI stacks installed:

```
tp wan -who eicon
tp lan -who bjames
```

CONAME - Connection Name

```
CONAME name [-DESC desc]
            [-FILTER filter]
            [-PARAMS params]
            [-POOL pool]
            [-TYPE TERM | PRINT]

[Gline parameter list]
```

Defines a connection name. e.g.

```
coname tssprod -desc "TSS production" -filter block_remote
-scid is40 -dmb tss       * Gline parameter list for all users
```

The CONAME parameters if used must be on the same line as the CONAME keyword. The optional list of Gline parameters must be on a single new line

-DESC adds a verbal description of the CONAME. It's currently unused except for documentation purposes. Maximum 30 characters.

-FILTER blocks access to this CONAME unless permitted by the filter.

-PARAMS points to PARAM directives that add to the line module parameter list.

-POOL points to POOL directives that add to the line module parameter list.

-TYPE is an optional identifier. If a CONAME is identified as TERM (terminal) or PRINT (printer) then *Ggate* (R6.1 and later) will supply the information to clients that request a CONAME list. The client can then display the available CONAMES in the appropriate drop-down lists, to make client configuration user-friendly. *Glink* (R7.2 and later) has this functionality.

Gline parameters

Gline parameters can be added to the CONAME directly, or via PARAMS or POOL directives. By default Gline parameters can be overridden by the client, but this can be inhibited using the '+' character as a prefix to the parameter in the directives in dsa.cfg. The client can be requested to ask for a parameter interactively using the '*' character for the parameter. If the '*' character is used as a prefix to a value, the value will be suggested as a default when the client asks for the parameter.

The set of line module parameters used to finally connect to the host is:

- The list from the CONAME directive
- If -PARAMS is specified this adds the options from **all** PARAMS directives with the specified name who's filters, if any, allows the client to access them. The order in which the options are added is that in which they are found in the dsa.cfg file
- If -POOL is specified it selects the **first** POOL directive in the dsa.cfg file with the specified name and a filter, if present, that allows the client to access it. The parameters from a random, as yet unused, line of parameters in the POOL directive are added
- Finally this set of parameters is overridden by any parameters from the user.

Note that the parameters from the user will not override any which have been inhibited using the '+' prefix. Note also that any RESTRICT that applies to the client will be applied to parameters supplied by the client, and may cause rejection of the connect attempt.

PARAMS - Parameters

```
PARAMS name [-FILTER filter]
Gline parameter list
```

Defines connection parameters. There may be multiple PARAMS directives with the same name, and all will be examined as candidates for addition to the parameter list for the client. If a filter is specified then only clients that are permitted by it get the parameter list added. Several CONAME directives can point to the same PARAMS directives.

Example, two users pick up different parameters from two PARAMS directives:

```
coname tssprod -desc "TSS production system" -params tssprod
   -scid is40 -dmb tss      * Gline parameter list for all users

params tssprod -filter john
   -usr john -pw calvin     * parameters for John only
params tssprod -filter sue
   -usr sue -pw hobbes      * parameters for Sue only
```

POOL - Parameter Pool

```
POOL name [-FILTER filter]
Gline parameter list
[Gline parameter list]
```

Defines a pool of connection parameters. There may be multiple `POOL` directives with the same name. They will be examined in the order in which they are found in `dsa.cfg`. If a filter is specified then only clients permitted by it can access the pool. The **first** `POOL` with the correct name and permission is selected. Several `CONAME` directives can point to the same `POOL` and the pool is then shared between clients connecting to the different `CONAME` directives. Each `Gline parameter list` line in the pool is independent of the others. They are allocated in round-robin order and the software is careful never to allocate the same line again as long as it is in use by a client. Typically the parameter lines will define unique LIDs to be shared by a group of clients.

Example: three pools of mailbox extensions for use as TP8 LIDs, one for users in Oslo, the second for users in Bergen and the third for users in Lillehammer:

```
coname tp8prod -desc "TP8 production system" -pool tp8prod
   -scid is40 -dmb tss      * Gline parameter list for all users

pool tp8prod -filter oslo
   -dx os01      * will be picked up by one Oslo user
   -dx os02      * will be picked up by one Oslo user
   -dx os03      * will be picked up by one Oslo user
   -dx os04      * will be picked up by one Oslo user

pool tp8prod -filter Bergen
   -dx be01      * will be picked up by one Bergen user
   -dx be02      * will be picked up by one Bergen user
   -dx be03      * will be picked up by one Bergen user
   -dx be04      * will be picked up by one Bergen user

pool tp8prod -filter lillehammer
   -dx li01      * will be picked up by one Lillehammer user


filter oslo
permit tcp 192.150.211.0/24      * Class C network
```

```
filter bergen
permit tcp 192.150.212.[00-10]    * macro for group
filter lillehammer
permit tcp 192.150.213.4          * single user
```

Macro facility

A simple macro facility eases configuration of parameters for large numbers of sessions. It expands numeric ranges only. Multiple macros on the same line must have the same range. You may use up to 8 macros on each line. e.g.

```
-mn mbx[01-03] -dx 00[12-14] -ur userrec[9-11]
```

expands to:

```
-mn mbx01 -dx 0012 -ur userrec9
-mn mbx02 -dx 0013 -ur userrec10
-mn mbx03 -dx 0014 -ur userrec11
```

The expansion will use the same number of digits if a numeric is written with leading zeroes i.e. `9-11` could have been written `09-11` to make the first expansion `userrec09`.

```
filter bergen
permit tcp 192.150.212.[00-03]    * macro for group
```

expands to:

```
filter bergen
permit tcp 192.150.212.00
permit tcp 192.150.212.01
permit tcp 192.150.212.02
permit tcp 192.150.212.03
```

COSTRICT - Connection Name Strictness

```
COSTRICT [-FILTER filter]
```

Defines strict handling of missing connection names. If a client connects and specifies an undefined connection name then the default is to simply ignore this and allow the client to continue and specify other parameters. If you specify `COSTRICT` then the default instead gives an error message. If you specify a filter then an error message is given only to those that are permitted by it.

RESTRICT - Parameter Restrictions

```
RESTRICT [-FILTER filter]
Gline parameter list
[Gline parameter list]
```

Defines Gline parameters that should be restricted from general use. If a filter is specified then only those that are permitted by it are subject to the restrictions. While this may seem backwards at first glance it was done this way so that all the directives can be read the same way, namely that the directive is visible only if the filter specified returns the client as permitted. You can specify an unlimited number of `RESTRICT` parameters and they will all be processed. Each `Gline parameter list` line within a `RESTRICT` is independent of the others. All Gline parameters in a `Gline parameter list` line need to be matched by options given by the client for the restriction to take effect. An asterisk (*) can be specified as a wild card to match any parameter value. Restrictions are only checked against parameters specified directly by clients, not against parameters specified in other parts of the `dsa.cfg` file.

Examples:

```
* Absolutely nobody is allowed to use his/her own SCID.
restrict
-scid *

* Those permitted through the remote_users filter are not allowed
* to use any pair of -scid or -dmb options, nor allowed to
* use the -ln option.
restrict -filter remote_users
-scid * -dmb *
-ln *

filter remote_users
* note that this filter allows all IP addresses
* so the restrict above applies to all remote users
permit TCP *
```

FILTER - Access Filter

```
FILTER name
PERMIT|DENY TCP|LOCAL|APPL|USER|NODE *|expression
...
```

Defines a filter for security screening. A filter is a list of one or more lines, each starting with PERMIT or DENY. Each line is checked in sequence until a match is found, then the permit/deny status of that line is returned. There's an implicit line to deny everything at the end of the definition. Local clients

running *Qsim*, *V78sim* etc. will match filter lines with LOCAL specified. Local clients are the *Host Links* products on the system that owns the /usr/gar (UNIX/Linux) or \gar (Windows) directory, or on other systems that mount it over the network. Clients coming in through *Ggate* will match filter lines with TCP specified. Any client will match filter lines with APPL specified.

TCP expression:

The expression is IPaddress[/bits]. The /bits part specifies how many bits of the IP-address will be used during matching. The default is 32, i.e. use the entire IP-address. If you want to permit/deny the whole of a class C network specify 24 bits. For the whole of a class B network specify 16 bits. For the whole of a class A network specify 8 bits. The asterisk (*) matches anything. Only clients coming into *Ggate* will match TCP filters.

LOCAL expression:

The expression is the system's UNIX/Linux or Windows communications node name. **NB!** it is not the DSA node name. This name can be displayed using the command uname -n on UNIX/Linux systems, and displayed in the Network section of the control panel on Windows server systems. Any application running on a system with this communications node name will match the expression. An asterisk (*) matches any communications node name. A specific expression is only needed when you filter applications running on one of several systems sharing a single dsa.cfg on a file server.

APPL expression:

The expression is one of Glink, Qsim, V78sim, Pthru, G3270, G5250, Padcon, Gmailer, Gspool, GUFT, GUFTSRV or Unknown.

```
filter block_remote
   permit local *                # implicitly denies remote
filter allow_ip
   deny tcp 192.150.211.4        # first deny one terminal
   permit tcp 192.150.211.0/24   # allow rest of CLASS C network
   permit tcp 18.0.0.0/8         # allow a class A network
   permit tcp 128.38.0.0/16      # allow a class B network
filter allow_qsim
   permit appl qsim              # applies to anyone using Qsim
```

USER expression:

The expression is the user name that is supplied by the client in the 'logon packet' (in the -us parameter). The user name is normally the user account name in the local client network. The asterisk (*) matches anything. Glink clients need version 6.1.4 or later. For Host Link applications, release 5.3 or later is required.

NODE expression:

The expression is the unique workstation ('node') name supplied by the client in the 'logon packet' (in the -st parameter). This filter type is suitable in configuration where the TCP expression cannot be used to identify a particular workstation i.e. if DHCP is used or in some firewall configurations (the IP address is generated dynamically). The workstation name is case sensitive. The asterisk (*) matches anything. Glink clients need version 6.1.4 or later. For Host Link applications, release 5.3 or later is required.

Use of filters to restrict access

In this example the class C IP networks 192.150.211.0, 193.71.16.0, 193.71.17.0, 193.71.18.0 and 193.71.19.0 should have access to both CONAMEs. All other IP addresses should only have access to the tp8test CONAME. Remember there is an implicit 'deny' of everything at the end of a filter.

```
# define a CONAME for a TP8 test system 'gartest' with no filters
# the coname is then available to anyone

coname tp8test -desc "TP8 Testapplication gartest" -pool tp8test
    -hm CXI -tm TXT 7801-da gartest -dn ph14

# define a coname for a mailbox '2wa-rno' with a filter allowing only
# the desired IP addresses

coname 2warno -desc "TP8 Testapplication 2wa rec no" -filter external
    -hm CXI -tm TXT7801 -da 2wa-rno -dx jimp -dn ph14

filter external
permit local *                      # allow Qsim, V78sim etc.
permit tcp 192.150.211.0/24         # allow class C network (use 24 bits)
permit tcp 193.71.16.0/22           # allow 16, 17, 18 and 19 (use 22 bits)
```

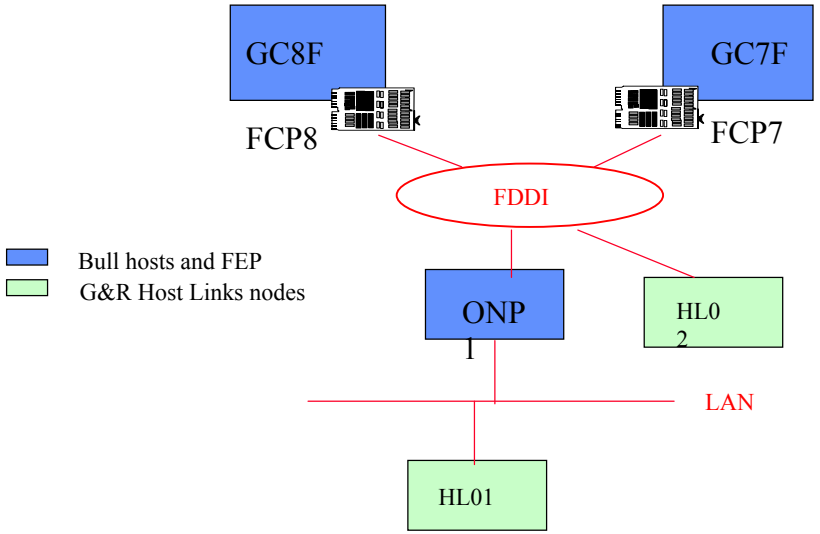
In this example all IP addresses except the ranges above should be prevented from specifying any node name or application name parameters directly. This is typically used to force access to happen through a CONAME.

```
# define a restrict parameter with a filter that allows through all
# TCP addresses which are to be restricted. The TCP addresses that are
# permitted by the filter meet the restriction
```

```
restrict -filter internal   # addresses passing the filter cannot choose
    -dn *                 # the default node to which they connect
    -da *                 # the application mailbox to which they connect
    -ln *                 # or to listen for incoming connects

filter internal
deny tcp 192.150.211.0/24   # they are not permitted by the filter
deny tcp 193.71.16.0/22     # and so DO NOT meet the restriction
permit tcp *                # rest are permitted, and meet the restriction
```

Sample dsa.cfg for RFC1006



Brief description of the configuration

HL01 - is a *Host Links* node with a LAN connection to a MainWay (ONP1). The ONP adapter in the MainWay supports RFC1006 and acts as a network switch that passes on the DSA session records to GC8F and GC7F. The MainWay sees HL01 as both a DSA and a DIWS node.

HL02 - is a *Host Links* node with an FDDI connection directly to the FCP7 adapter of GC7F and the FCP8 adapter of GC8F. GC7F and GC8F see HL02 as both a DSA and a DIWS node.

GC8F - is a DPS9000 system with an FDDI adapter (FCP8). This is accessed directly from the **HL02** *Host Links* system (i.e. without going through a MainWay) for optimal speed and performance.

GC7F - is a DPS7000 with an FDDI adapter (FCP7). This is accessed directly from the **HL02** *Host Links* system (i.e. without going through a MainWay) for optimal speed and performance.

ONP1 - is a Bull MainWay that is used for network access from the **HL01** *Host Links* system to GC8F and GC7F.

The Host Links node 'HL01'

Windows Server	\gar\config\dsa.cfg
UNIX/Linux	/usr/gar/config/dsa.cfg

```
# Local Host Links DSA node HL01
#
sc hl01 -addr 49:001

# local G&R RFC1006 transport provider (IP=202.1.1.1)
tp rfc -who gar

#
# Remote DPS7000, DSA node GC7F thru LAN connected ONP1
#
rsc gc7f -addr 49:007 -ts onp1

#
# Remote DPS9000, DSA node GC8F thru LAN connected ONP1
#
rsc gc8f -addr 49:008 -ts onp1

# transport station route (Mainway)
ts  onp1 -class 0 -ns 202.1.1.5 -tp rfc
```

The Host Links node 'HL02'

Windows Server	\gar\config\dsa.cfg
UNIX/Linux	/usr/gar/config/dsa.cfg

```
#
# Local Host Links DSA node HL02
#
sc hl02 -addr 49:002

# local G&R RFC1006 transport provider (IP=202.1.1.2)
tp rfc -who gar

#
# Remote DPS7000, DSA node GC7F thru FCP7
#
rsc gc7f -addr 49:007 -ts fcp7
# FCP7 transport station
ts  fcp7 -class 0 -ns 202.1.1.7 -tp rfc

#
# Remote DPS9000, DSA node GC8F thru FCP8
#
rsc gc8f -addr 49:008 -ts fcp8
# FCP8 transport station
ts  fcp8 -class 0 -ns 202.1.1.8 -tp rfc
```

The DPS9000 node ‘GC8F’

Osi.cfg

```
STACK I_NOERRFLAG
NETSHARE
TCIVMO NAME=IVMO8 TPUSIZE=1024
NRSC NAME=HL01 NRTS=HL01TS
NRTS NAME=HL01TS TYPE=INET IP_ADDR=202.1.2.1/102
NRSC NAME=HL02 NRTS=HL02TS
NRTS NAME=HL02TS TYPE=INET IP_ADDR=202.1.2.2/102
```

Routes.wdf

```
create_node_route_list hl01 -nr1 fe4
create_node_route_list hl02 -nr1 fe4
```

Loadscen

```
sti W3NX00T20010
macaddr 080038100FE8
```

The DPS7000 node ‘GC7F’

Netgen’

```
SYS  GC7F PF=LSYS SCID=49:007 ISL=(54-43-55,EA81) OBJLIST=1,TSEL=47433746,
          NSAP=39250F034500010800381OOFE7;

COMM 'NCC1',
SVR  NAME=NCC1 TYPE=OCS LCT=EA71 BUFNB=(32768,32768) BUFSZ=256;
LCT  NAME=EA71 TYPE=NCC IADDR=08-00-38-10-0F-E7 MAXCN=5000 WATCH=1200;

COMM 'REMOTE HOST LINKS SYSTEMS';
RSYS NAME=HL01, PF='STID/ISO/SID3', RSC=HL01;
RSC  NAME=HL01, ADDR=DSA, SCID=49:001, SVR=NCC1;
RSYS NAME=HL02, PF='STID/ISO/SID3', RSC=HL02;
RSC  NAME=HL02, ADDR=DSA, SCID=49:002, SVR=NCC1;

COMM 'MAINWAY';
RSYS NAME=ONP1, PF=MW2000, RSC=ONP1; MW2000=’MW2000/DNS-E/V1’
RSC  NAME=ONP1, ADDR=DSA, SCID=49:005, SVR=NCC1;
```

FCP7

```
STACK TSNAME=NCC1
TCIVMO NAME=IVMO1 TPUSIZE=1024
LTS_INET NAME=LTSNEW IP_ADD=202.1.1.7 PORT=102 MAXTC=4500 TPUSIZE=4096
NRSC NAME=HL01 NRTS=HL01NCC1
NRTS NAME=HL01NCC1 TYPE=INET  IP_ADD=202.1.1.1 PORT=102
NRSC NAME=HL02 NRTS=HL02NCC1
NRTS NAME=HL02NCC1 TYPE=INET  IP_ADD=202.1.1.2 PORT=102
```

The MainWay

```
&
&    LOCAL MAINWAY NODE 'ONP1', ADDR=49:005
&
SC ONP1 LOC -ADDR 49:005 -ISOPLG
TS ONP1 LOC -ADDR 49:005 -NET 492008003811008D -CNX 900  -SERV CONS

&
&   DPS7000, SC RMT GC7F   ADDR=49:007
&
SC GC7F RMT  -ADDR 49:007 -SR GC7F -NAT DSA
SR GC7F ISO  -TS GC7F
TS GC7F DIWS -NR GC7F
NR GC7F LAN1 -PL GC7F
PL GC7F CSM1 -CB ETH1 -ETHAD 080038100FE7

&
&   DPS9000, SC RMT GC8F   ADDR=49:008
&
SC GC8F RMT  -ADDR 49:008 -SR GC8F -NAT DSA
SR GC8F ISO  -TS GC8F
TS GC8F DIWS -NR GC8F
NR GC8F LAN1 -PL GC8F
PL GC8F CSM1 -CB ETH1 -ETHAD 080038100FE8

&
&   Host Links node SC RMT HL01, 49:001
&
SC HL01 RMT -ADDR 49:001 -SR HR01
&
SR HR01 IPS -TS HR01
TS HR01 IPS -IPADD 202.1.1.1 -CT CTIP
&
NR NRIP IPS -CT CTIP
LL LLIP IPS -CT CTIP
CT CTIP IPS
&
```

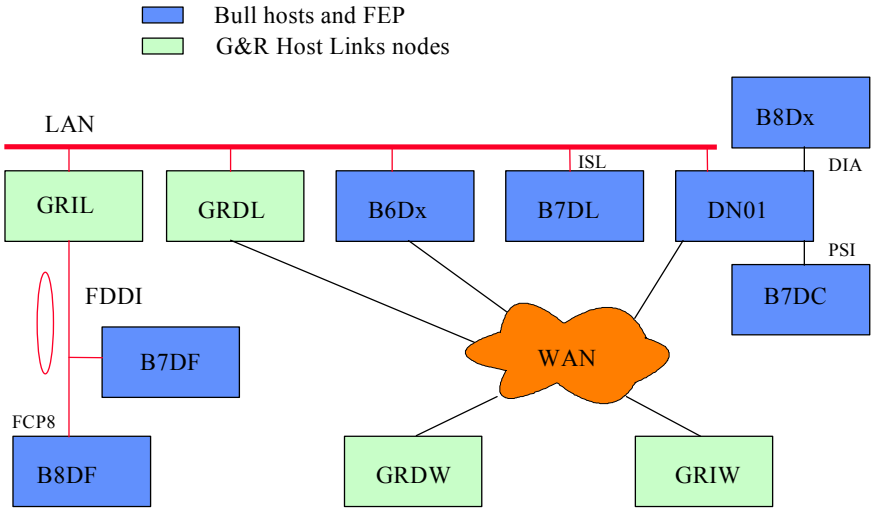
Sample dsa.cfg for OSI Transport

This sample configuration is meant to cover many of the most used configuration scenarios and will hopefully be useful when setting up new configurations or modifying existing ones. It shows various types of Bull mainframe systems and some *G&R Host Links* nodes. Both LAN and WAN configurations are included in addition to a simple example of an FDDI link.

The various configuration files do not necessarily include all of the other accessible systems, but most combinations of interconnection are included.

Some of the mainframe configuration examples are not complete, but the names and addresses used in the primary network are included. Although many of the examples have been used on the systems in question, differences in configuration syntax on different software releases should be expected and a copy of the necessary configuration manuals is essential when building a configuration.

Sample configuration



Brief description of the configuration

For simplicity all GCOS6 hosts are referred to as DPS6s, all GCOS7 hosts as DPS7s and all GCOS8 hosts as DPS8s except for the ones with FCP8 adapters, which are referred to as DPS9000s:

GRDL - is a *Host Links* node with both LAN and WAN connection. It is configured to reach a DPS7 over LAN (ISL) and a DPS8 over WAN (through the FEP). It can be seen as both a DSA and a DIWS node by the mainframes.

GRIL - is a *Host Links* node connected to the LAN but also configured to access a DPS9000 over FDDI (directly through the FCP8 controller). Over the LAN it can access the DPS6, the DPS7 (directly on ISL) and the DPS8 via the LAN adapter of the FEP.

GRIW - is a *Host Links* node with X25 access to a DPS7 and a DPS8 via the FEP. It is configured as a DIWS node in the host configurations.

GRDW - is a *Host Links* node with X25 access to a DPS6, a DPS7 and a DPS8. It is configured as a DSA node in the host configurations.

B8DF - is a DPS9000 system with an FDDI adapter (FCP8). This is accessed directly from the **GRIL** *Host Links* system (i.e. without going through a MainWay) for optimal speed and performance.

B6Dx - is a DPS6 with both WAN and LAN capabilities. It has 3 separate session control entities defined, but the most important one here (as far as terminal sessions is concerned), is **B6DL** (the SNM session control name).

B7DL - is a DPS7 with a MicroFEP and an ISL. The *Host Links* nodes are configured to reach this system directly over LAN (i.e. the ISL) without passing though the MicroFEP.

B7DF - is a DPS7 with an FCP7 (FDDI controller). The *Host Links* nodes are configured to reach this system directly over FDDI.

DN01 - is a Bull FEP (Datanet) that is used here for network access from the various *Host Links* nodes to DPS8 and DPS7.

B8Dx - is a Bull DPS8. Two session control names are defined; **B8DT** for access to DNET/ROUT applications (TSS, DMIV/TP etc) and **B8DC** for CXI applications (TP8 etc).

The configuration files

The Host Links node 'GRDL'

Windows Server	\gar\config\dsa.cfg
UNIX/Linux	/usr/gar/config/dsa.cfg

```
# dsa.cfg: Configuration for G&R Host Links node GRDL
# Local node. DSA/DIWS type, LAN access and WAN access.
sc grdl -addr 54:60

# Remote nodes accessed by GRDL
# Another Host Links system (GRIL)
rsc gril -addr 54:61 -ts gril_lan
ts  gril_lan -class 4 -ns 02608c2d8023 -tp grdl_lan

# A DPS7 system, B7DL, LAN access (over ISL)
rsc b7dl -addr 1:4 -ts b7dl_lan
ts  b7dl_lan -class 4 -ns 080038017777 -tp grdl_lan

# The DPS8 system (DNET/ROUT and CXI), accessed over WAN in DN01
rsc b8dt -addr 1:1 -ts dn01_wan
rsc b8dc -addr 1:2 -ts dn01_wan

# The Datanet, used for WAN access to DPS8
ts dn01_wan -class 2 -ns 130399 -tp grdl_wan

# Local transport providers (TP4 and TP2)
tp grdl_lan -attach 02608c200441
tp grdl_wan -attach 130355
```

The Host Links node 'GRIL'

Windows server	\gar\config\dsa.cfg
UNIX/Linux	/usr/gar/config/dsa.cfg

```
# dsa.cfg:  Configuration for G&R Host Links node GRIL
# Local node definition. DIWS type for LAN access, DSA used in FDDI
# case. The DSA address is not given here (only necessary for
# DSA200 type of sessions)
sc gril

# Remote nodes accessed by GRIL
# The DPS6 system (SNM and TPI)
rsc b6dl -ts b6dl_lan
rsc b6dt -ts b6dl_lan
ts  b6dl_lan -class 4 -ns 080038363E18 -tp gril_lan

# Another G&R Host Links system, GRDL. DIWS type, LAN access.
rsc grdl -ts grdl_lan
ts grdl_lan -class 4 -ns 02608c200441-tp gril_lan

# A DPS7 system, B7DL, LAN access (over ISL)
rsc b7dl -ts b7dl_lan
ts  b7dl_lan -class 4 -ns 080038017777 -tp gril_lan

# A DPS8 system, B8DC (CXI), LAN access over DN01
```

```
rsc b8dc -ts dn01_lan
ts  dn01_lan -class 4 -ns 080038200557 -tp gril_lan

# A DPS8 system, B8DF, direct FDDI access to FCP8
rsc b8df -ts b8df_fddi

#   NB! the FDDI MAC address (10001c08082a) must be specified
#   in SMT format. The corresponding MAC LLC address is: 080038101054
#   For a description of the conversion, see the Gline documentation
ts  b8df_fddi -class 4 -ns 10001c08082a -tp gril_fddi

# Another  DPS7 system, B7DF, FDDI
rsc b7df -ts b7df_fddi
ts  b7df_fddi -class 4 -ns 10001c08f0ee -tp gril_lan
#   Re NS; the corresponding MAC LLC address is: 080038100f77

# Local transport provider (TP4)

tp gril_lan  -attach 02608c2d8023
tp gril_fddi -attach 10005ab83588
```

The Host Links node 'GRDW'

Windows server	\gar\config\dsa.cfg
UNIX/Linux	/usr/gar/config/dsa.cfg

```
# dsa.cfg: Configuration for G&R Host Links node GRDW
# Local node. DSA/DIWS type, WAN access only.
sc grdw -addr 54:58

# Remote nodes accessed by GRDW
# The DPS6 system (SNM), accessed over X25
#     b6dl -addr not necessary, SNM uses DSA300
rsc b6dl -ts b6dl_wan
ts  b6dl_wan -class 2 -ns 130405 -tp grdw_wan

# The DPS7 system, accessed over X25 in DN01
rsc b7dc -addr 1:3 -ts dn01_wan

# The DPS8 system (DNET/ROUT and CXI), accessed over X25 in DN01
rsc b8dt -addr 1:1 -ts dn01_wan
rsc b8dc -addr 1:2 -ts dn01_wan

# The Datanet, used for WAN access to DPS7 and DPS8
ts  dn01_wan -class 2 -ns 130399 -tp grdw_wan

# Local transport provider (TP2, local x25 number 130366)
tp grdw_wan -attach 130366
```

The Host Links node 'GRIW'

Windows NT/2000	\gar\config\dsa.cfg
UNIX	/usr/gar/config/dsa.cfg

```
# dsa.cfg: Configuration for G&R Host Links node GRIW
# Local node. DIWS type, WAN access only.
sc griw

# Remote nodes accessed by GRIW
# The DPS7 system, accessed over X25 in DN01
rsc b7dc -ts dn01_wan

# The DPS8 system (DNET/ROUT), accessed over X25 in DN01
rsc b8dt -ts dn01_wan
# The Datanet, used for WAN access to DPS7 and DPS8
ts  dn01_wan -class 2 -ns 130399 -tp griw_wan

# Another Host Links system GRDW
rsc grdw -ts grdw_wan
ts  grdw_wan -class 2 -ns 130366 -tp griw_wan

# Local transport providers (TP2)
tp griw_wan -attach 130367
```

The DPS6 node 'B6Dx'

```
>WDD>STARTUP>NW_CONFIG:

* Local node B6DL
NODE      -NAME B6DL -PSN  1 -AREA 1 -TSN 5       -- Local node
SC        -Q_CAP 8 -LETTERS 400                   -- Max. no of letters

SCA       -NAME B6DL -TYPE TDSA                   -- SNM DSA session type
SCA       -NAME B6DT -TYPE DSS  -TS_ID X'0105'    -- TPI session type
SCA       -NAME B6DD -TYPE D6SC -TS_ID X'0106'    -- DSA session type

ISOS      -NAME ISOS                              -- enables ISO sess.
TC        -LC_CAP 40 -TU_SIZE 1497 -CREDIT 3 -RETRY  3
          -T1 10 -T2 60 -T3 5 -T4 5 -T5 2

ISOTC     -NAME ISOT -MAX_REFS 20

LTSEL     -DEFAULT -UPPER_TYPE SWIT

NSAP      -NAME NSB6DL -OWNER DSA                 -- used by DSA transport

NSAP      -NAME NSB6DLI -OWNER ISOT               -- used by ISO

SNSAP     -NAME SNB6DL -NTW TELENET -MAXVC 12 -LCNT 1024 1035
          -SIZE 128 128 -WINDOW 2 2 -FC -CALL 130405 -LOCAL_SAP LL10

LL        -NAME LL10 -OWNER X25 -PL PL10 -ADDR A  -- logical link
          -FRL 133 -T1  3 -WINDOW 7

PL        -NAME PL10 -LRN 10 -TYPE HDLC -SPD 96   -- physical link

* Secondary network                                      --
```

```
TERM      -NAME SNM150 -MODE USER  -NTD NTDVIP -TYPE DKU7107 -EL 150
TERM      -NAME ROP150 -MODE IMPLO -NTD NTDVIP -TYPE TTU8125 -EL 150
RTERM     -NAME PRX150 -CM DSA -AD1 GRDL -AD2 ROPGRDL -GN ROP150
TERM      -NAME SNM151 -MODE USER -NTD NTDVIP -TYPE VIP7804
          -EL 151,152,153,154,155,156,157,158,159 -- LRNs from clm_user

* Remote nodes                                                --

RTSEL     -DEFAULT -CLASS 4 -TU_SIZE 1024        -- if no matching RTSEL

* G&R Host Links node GRDL, LAN access, DSA type

SRISO     -RNSAP_NAME   NSGRDLI
RSCA      -NAME         GRDL    -TYPE D6SC
RTSEL     -RNSAP_NAME   NSGRDLI -T_SEL -HEX 40024752444C -TU_SIZE 1024
          -NO_CHECKSUM -CLASS 4

RNSAP     -NAME         NSGRDLI -OWNER ISOT -NSAP NSB6DLI
ROUTE     -NAME         RTGRDLI -LOCAL_SAP LPTP18 -REMOTE_SAP GRDLSAP
          -SUBLAYER     LACS -PROTOCOL CLNP -NULL
LLAN      -L_USER       LPTP18  -R_USER GRDLSAP -FRAME 1497

* Another G&R Host Links node GRIL, LAN access, DIWS type

SRISO    -RNSAP_NAME   NSGRILI
RSCA     -NAME         GRIL    -TYPE IDWS -TSAP_MODE IDS -IMPL_SES DSS
RTSEL    -RNSAP_NAME   NSGRILI -T_SEL -HEX 40014753494C -TU_SIZE 1024
         -NO_CHECKSUM -CLASS 4
RNSAP    -NAME         NSGRILI -OWNER ISOT -NSAP NSB6DLI
ROUTE    -NAME         RTGRILI -LOCAL_SAP LPTP18 -REMOTE_SAP GRILSAP
         -SUBLAYER LACS -PROTOCOL CLNP -NULL
LLAN     -L_USER       LPTP18  -R_USER GRILSAP -FRAME 1497

* Yet another G&R Host Links node GRDW, X25 access, DSA type

SRISO    -RNSAP_NAME   NSGRDWI
RSCA     -NAME         GRDW    -TYPE D6SC
RTSEL    -RNSAP_NAME   NSGRDWI -T_SEL -HEX 400247534457 -TU_SIZE 2048
         -NO_CHECKSUM -CLASS 2
RNSAP    -NAME         NSGRDWI -OWNER ISOT -NSAP NSB6DLI
ROUTE    -NAME         RTGRDWI -LOCAL_SAP SNGRDL -REMOTE_SAP SNGRDWI
•      SUBLAYER X25 -PROTOCOL X25 -ENCODING TP
RSNSAP   -NAME         SNGRDWI -CALL 130366

* Network Administration, left out of example


LANC-F :


*
* DPS6 'lanc-f' directives for B6DL, DNTD and DSA SAPs
*
GRPSAP GNTD,XPT18/LNTD18
*
TLIN XPT18,CLS4,X'1800',0
LCLSAP LNTD18,LLC18/LXPT18,X'0002',CONN=16,MAXPDU=1024
*
LLIN LLC18,TYP1,X'1800',0
*
```

```
* various local SAPs
*
LCLSAP LXPT18,,X'F8'/X'080038363E18'
LCLSAP LDDS18,,X'14'/X'080038363E18'
LCLSAP LTNV18,,X'28'/X'080038363E18'
LCLSAP LPTP18,,X'20'/X'080038363E18'
*
* GRDL SAP and MAC address
*
REMSAP GRDLSAP,,X'20'/X'02608C200441'
*
* GRIL ditto
*
REMSAP GRILSAP,,X'20'/X'02608c2d8023'
*
* local physical line
*
PLIN PHY18,CSMA,X'1800',0
LCLSAP LMAC18,,X'080038363E18'
```

The DPS7 node 'B7DL'

```
COMM  'Sample DPS7 config (extract)'
COMM  'B7DL DPS7000 local definitions'
SYS    NAME=B7DL PF=LSYS SCID=1:4 ISL1=(08-00-38-01-77-77 EA01 CB01 )
       TCPIP=1 OBJLIST;

COMM  'MicroFEP';
RSYS   B7DM PF='CNP7/CNS7/A2' RSC=B7DM RCT=RCT_B7DM;
RSC    B7DM SCID=1:10 SVR=TNS RTS=RTS_B7DM;
NR     NR_B7DM CP=CP_EA01 RPL=RPL_B7DM TP=TP_B7DM;
RPL    RPL_B7DM IADDR=08-00-38-33-00-02 CB=CB01 RCT=RCT_B7DM;
RCT    RCT_B7DM;
RTS    RTS_B7DM NR=NR_B7DM;
TP     TP_B7DM;

COMM  'G&R Host Links node GRDL, access over ISL'
SYS    NAME=GRDL PF='STID/ISO/SID4' SCID=54:60 ISL1=(02-60-8C-20-04-41 CB01)
OBJLIST;
COMM  'DPS6 system B6DT, access over MicroFEP'
RSYS   NAME=B6DT,PROFILE='DPS6/DSS/3.1',RSC=B6DT;
RSC    NAME=B6DT,SCID=1:5,SERVER=TNS,PROTOCOL=DSA200,RTS=RTS_B7DM;
```

The DPS9000 node 'B8DF'

osi.cfg:

```
#
# Sample FCP8 config (osi.cfg) for B8DF, direct FDDI access to G&R DSA node
#
stack i_noerrflag esis
tcivmo name=ivmo_fddi tpdusize=4096 maxrcrdt=7 maxretry=6
#
# G&R node 'GRIL' (Escala box)
#
nrsc name=gril nrts=grilts proto=dsa
nrts name=grilts type=diws ivmo=ivmo_fddi macadr=10005ab83588
```

routes.wdf:

```
#
# Local session control for 'B8DF, channel fe4 (in routes.wdf)
#
create_node_route_list gril -nrl fe4
```

loadscen:

```
#
# FCP8 MAC (LLC) Ethernet address (in loadscen file)
#
sti W3NX00T20010
macadr 080038101054
```

The Datanet node 'DN01'

```
&************************************************************
&  G&R Host Links DSA node 'GRDW'. WAN access
&************************************************************
SC GRDW RMT    -SR GRDW -ADDR 54:58 -NAT DSA
SR GRDW ISO    -TS GRDW
TS GRDW DIWS   -NR GRDW -CLASS 2 -CRRE 15 -FEN 13 &
               -TPDU 512 -NSDU 512 -RTRY 3 -T1 40 -T2 600
NR GRDW SW     -RMT GRDW -NS DPAK
NS GRDW RMT    -CALL 130366

&************************************************************
&  G&R Host Links DIWS node 'GRIW'. WAN access
&************************************************************
SC GRIW RMT    -SR GRIW -ADDR 54:59 -NAT ISO
SR GRIW ISO    -TS GRIW
TS GRIW DIWS   -NR GRIW -CLASS 2 -CRRE 15 -FEN 13 &
               -TPDU 512 -NSDU 512 -RTRY 3 -T1 40 -T2 600
NR GRIW SW     -RMT GRIW -NS DPAK
NS GRIW RMT    -CALL 130367

&************************************************************
&   G&R host Links DSA node 'GRDL'. LAN access
&  (GRDL here configured using 2 different names (TSAPs)
&   but using the same 'route')
&************************************************************
SC GRDL RMT    -SR GRDL -ADDR 54:60 -NAT DSA
SC GRD2 RMT    -SR GRD2 -ADDR 54:70 -NAT DSA
SR GRDL ISO    -TS GRDL
SR GRD2 ISO    -TS GRDL
TS GRDL DIWS   -NR GRDL -CLASS 4 -CRRE 15 -FEN 13 &
               -TPDU 1024 -NSDU 1024 -RTRY 5 -T1 400 -T2 1600
NR GRDL LAN1   -PL GRDL
PL GRDL CSM1   -ETHAD 02608c200441 -CB ELA1
&************************************************
&  DPS9000 TSS with channel access
&************************************************
SC B8DT RMT    -ADDR 1:1 -SR B8DT -NAT DSA -DFLT -BRK ATPGLT
SR B8DT L66G   -CH B8DT
CH B8DT L66G   -LNNB 300 -CT DIA1 -T1 0 -ICML 750
CT DIA1 DIA    -PHAD X'0800'
&************************************************
& DPS9000 CXI with channel access
&************************************************
```

```
SC B8DC RMT    -ADDR 1:2 -SR CXIA
SR B8DC CXI    -CH B8DC
CH B8DC CXI    -CT DIA1 -T1 6000 -T2 2400 -CNX 1000

&************************************************
& DPS7000 with channel access
&************************************************
SC B7DC RMT    -ADDR 1:3 -BRK AT -SR B7DC
SR B7DC GC64   -CH B7DC
CH B7DC GC64   -T3 3000 -RCH 8 -CNX 200 -CT PSI1
CT PSI1 PSI    -PHAD X'0C00' -MXCH 16

&************************************************
& Local LAN connection
&************************************************
CB ELA1 LAN1   -PL ELAN
PL ELAN CSMA   -CT ELAN -MXSZ 1518 -PHAD X'8000' -ETHAD 080038200557
CT ELAN ELNC   -PHAD X'8000'

&************************************************
& Local X.25
&************************************************
NS DPAK X25    -CALL 130399 -NBVC 1034 -LL DPAK -NTW TELENET
LL DPAK BDL    -PL DPAK -LAPB -FRL 133 -T1 100 -W 7
PL DPAK HDLC   -PHAD X'1600' -PHTP 2140 -CT MLC2 -SPD 9600
```

The DPS7 node 'B7DC'

```
COMM       'B7DC DPS7 local definitions'
LSYS       NAME=B7DC,PROFILE='DPS7/GCOS-7/V3',LSC=B7DC;
LSC        NAME=B7DC,SCID=1:3;

COMM       'Remote systems'
COMM       'G&R - Host Links GRDW, DSA node';
RSYS       NAME=GRDW,PROFILE='STID/ISO/SID3',RSC=GRDW;
RSC        NAME=GRDW, ADDR=DSA, SCID=54:58,SERVER=DTN1;

COMM       'G&R - Host Links GRIW, DIWS node';
RSYS       NAME=GRIW,PROFILE='STID/ISO/SID4',RSC=GRIW;
RSC        NAME=GRIW, ADDR=DSA, SCID=54:59,SERVER=DTN1,PRTC=SID;
```

The DPS7 node 'B7DF'

```
COMM       'Sample DPS7 with FCP7 access'

SYS        NAME=B7DF,PF=LSYS,SCID=01:08,NCC1=(08-00-38-10-0F-77,EA01);

COMM       'Host Links node GRIL, DIWS type';

RSYS       NAME=GRIL,PF='STID/ISO/SID3',RSC = GRIL;
RSC        NAME=GRIL,ADDR=DSA,SCID=54:061,SVR=NCC1, PRTC=SID;
```

The corresponding FCP7 configuration file:

```
STACK TSNAME=NCC1
TCIVMO NAME=IVMO2 TPUSIZE=1024 MAXRCRDT=7 MAXSCRDT=7 MAXRETRY=3
T1TIME=10000 WTIME=60000 ITIME=180000  ACKTIME=4000
NRSC NAME=GRIL NRTS=GRIL
NRTS NAME=GRIL TYPE=DIWS MACADR=08005a1dac11 IVMO=IVMO2
```

A shared dsa.cfg for all the Host Links nodes

In some situations it makes sense to build and maintain one `dsa.cfg` file for all the *`Host Links`* nodes in the network. If two or more *`Host Links`* systems share a `dsa.cfg` file in a shared file system you must use the `-UNAME` directive to assign parameter values to the correct Host Links system. The `-UNAME` parameter can also be useful if you prefer to maintain a single copy of the `dsa.cfg` file and simply distribute it to all *`Host Links`* systems when it has been modified. The –UNAME parameter refers to the UNIX or Windows server system's communications node name. **NB!** It is not the DSA node name. This name can be displayed using the command `uname -n` on UNIX/Linux systems, and displayed in the Network section of the control panel on Windows server systems.

Windows Server	`\gar\config\dsa.cfg`
UNIX/Linux	`/usr/gar/config/dsa.cfg`

```
#
# Shared dsa.cfg file for all Host Links systems in the example
#

#
#  Host Links system (local definitions)
#

#  DSA Node GRIL, communications node name 'Tor',  DIWS node
sc gril -uname Tor
tp lan  -uname Tor -attach 02608c2d8023
tp fddi -uname Tor -attach 10005ab83588

#  DSA Node GRIW, communications node name 'Hugin', DIWS node
sc griw -uname Hugin
tp wan  -uname Hugin -attach 130367

#  DSA Node GRDL, communications node name 'Munin', DSA node
sc grdl -addr 54:60 -uname Munin
tp lan  -uname Munin -attach 02608c200441
tp wan  -uname Munin -attach 130355

#  DSA Node GRDW, communications node name 'Sleipner', DSA node
sc grdw -addr 54:58 -uname Sleipner
tp wan  -uname Sleipner

#
#  Remote system (WAN access)
#

#  The Dps6 system (SNM)
rsc b6dl -ts b6dl_wan
ts  b6dl_wan -class 2 -ns 130405 -tp wan

#  The Dps7 system, accessed over X25 in DN01
rsc b7dc -addr 1:3 -ts dn01_wan
```

```
#  The Dps8 system (DNET/ROUT and CXI), accessed over X25 in DN01
rsc b8dt -addr 1:1 -ts dn01_wan
rsc b8dc -ts dn01_wan

#  The DN, used for WAN access to Dps7 and Dps8
ts dn01_wan -class 2 -ns 130399 -tp wan

#  Host Links systems
rsc grdw -addr 54:58 -ts grdw_wan
ts  grdw_wan -class 2 -ns 130366 -tp wan
rsc griw -ts griw_wan
ts  griw_wan -class 2 -ns 130367 -tp wan

#  Remote systems accessed over LAN
#  Various G&R Host Links systems
rsc grdl -addr 54:60 -ts grdl_lan
ts  grdl_lan -class 4 -ns 02608c200441 -tp lan
rsc gril -ts gril_lan
ts  gril_lan -class 4 -ns 02608c2d8023 -tp lan

#  A DPS6 system, Session control addresses for SNM, TPI and DSA
rsc b6dl -ts b6dl_lan
rsc b6dt -ts b6dl_lan
rsc b6dd -ts b6dl_lan
ts  b6dl_lan -class 4 -ns 080038363E18 -tp lan

#  A DPS7 system, B7DL, over ISL
rsc b7dl -addr 1:4 -ts b7dl_lan
ts  b7dl_lan -class 4 -ns 080038017777 -tp lan

# A DPS8 system, B8DF, direct FDDI access to FCP8
rsc b8df -addr 1:9 -ts b8df_fddi
ts  b8df_fddi -class 4 -ns 10001c08082a -tp fddi


# Coname 'b8dcacce' used for B8DC access, pooled mailbox extensions
coname b8dcacce -desc "Pool for B8DC access" -pool lidpool
       -dn b8dc -da tp88 -hm cxi -tm vip7804

pool lidpool
•      -dx lid1
•      -dx lid2
•      -dx lid3
•      -dx lid4
•      -dx lid5
•      -dx lid6
•      -dx lidt

# Coname 'obsprt' referenced by -pco parameter for incoming print
coname obsprt -desc "Print from TDS"
       -ln -tm ttu8126 -mn mbx1
```

Appendix: Host Links Manuals

Below you find a complete list of all available Host Links manuals:

Installation	
Host Links	Installation and Configuration on UNIX/Linux
Host Links	Installation and Configuration on Windows
Line handling	
Gline	Line Handler and DSA/OSI Configuration
Ggate	Transparent Gateway
Gproxy	Network Manager & SNMP Proxy Agent
G&R SSL	Using SSL for security in G&R products
GlAPI	Application Programming Interfaces
Emulations	
Gspool	Network Printer Emulation
GUFT	Unified File Transfer
G3270	Emulating IBM 3270 Terminals
G5250	Emulating IBM 5250 Terminals
Pthru	Gateway to the Bull Primary Network
Qsim	Emulating Questar DKU7107-7211 & VIP7700-7760
V78sim	Emulating VIP7801 & VIP7814
Gweb	Web Browser Front-end for DKU, VIP7700-7760, VIP7800, IBM3270 and IBM5250 Emulations

Appendix: DSA test utilities

The Gline package includes a set of Gline communication utilities. These are used when testing and debugging connection problems. The utilities are delivered as part of the Gline package and can be used without any additional configuration. The nodes to be tested must of course be configured in the `dsa.cfg` file.

Gerror

Show the text message associated with a DSA reason code. Only the most common codes are supported i.e. the ones related to network, transport and session communication layers. Errors generated by the OSI-stack on the Host Links platform are not covered by this utility; please refer to the documentation from the vendor of the stack.

Example:

```
gerror 0109
Reporting component: Session control (01) 0109, Dialog
protocol error or negotiation failed (wrong logical
record).
```

For a detailed description of all reason codes, please consult the Bull manual ***OSI/DSA Network System Messages and Return codes*** (39A2 26DM).

Glnode

List and verify the communications parameters of the local node.

Example:

```
glnode
Local node name : GRDL
Local session control id : GRDL
DSA200 address (area:tsm): 54:60 (36:3C)
```

Gmacfix

When connecting to the FCP cards on the Bull mainframes via an Ethernet port on the LAN-Extender the address on the host side is given in the Ethernet (LLC) format. If you connect to an FDDI adapter directly or by an FDDI port of a LAN Extender, you must convert the MAC address to SMT. This utility can be used to convert the address format. Typically Bull LLC MAC addresses start with `080038` and the FDDI address starts with `10001C`.

Example:

```
gmacfix 080038000fab
MAC address 080038000fab = 10001c00f0d5
```

Gping

Connects to a remote system using the Gline parameters set on the command line. If successful it returns 'connected to application', otherwise it shows the error code returned.

Example:

```
gping -li dsa -dn b7dl -da iof -du jim -pw mydogsname
Gping - $$DSA: Connected to application
```

Grnode

Return the configured parameters (in `dsa.cfg`) and the state of a remote node.

Example:

```
grnode b6dl
Checking 'dsa.cfg' for node 'b6dl'
Session control id : B6DL
DSA200 address (area:tsm) : 1:5 (1:5)
Inactivity interval : 0
Route 0
Load balance percentage : 0
TP class : 2
TP expedited : 0
TPDU size : 0
Network address : 130405
```

Gtrace

Same as `gping` but writes the DSA/DIWS communication trace on the user's terminal (applicable to UNIX versions)

Example:

```
gtrace -li dsa -dn ln40 -da snm151
D6:Application event @ 14:17:17.6003. tokenitem = 00
D6:Application event @ 14:17:17.6082. tokenitem = 00
D6:Connect request called, node = LN40
D6:OurBufferSizes. ApplMaxXmit = 511, ApplMaxRecv = 500
Rec:4000 0002 s:2
Rec:506B 0010 s:16
etc etc
Gtrace - line trace ending.
Gtrace - $$DSA: Connected to application.
```

Gtsupd

Update the state of a transport route. Transport routes can be set automatically in a disabled state if a backup route is configured. When such a state change occurs the route will be set back to the enabled state after a configurable timer has expired. The default is 15 minutes. You can reset the state of such a route with the `gtsupd` utility.

Example:

```
gtsupd hipp -st enbl
TS-entry 'hipp' updated OK. Old state = LOCK, new state =
ENBL
```

Appendix: Error Codes

OSI/DSA error codes

Below is a list of OSI/DSA error codes and the corresponding description. These are the same descriptions that the `G&R/Gerror` utility will display when given the DSA code as a parameter.

code	Description
00xx	**General Errors**
0001	Open Failure in LC - Reject for unknown reason
0002	Open Failure in LC - Acceptor customer node inoperable
0003	Open Failure in LC - Acceptor customer node saturated.
0004	Open Failure in LC - Acceptor mailbox unknown.
0005	Open Failure in LC - Acceptor mailbox inoperable.
0006	Open Failure in LC - Acceptor mailbox saturated.
0007	Open Failure in LC - Acceptor application program saturated
0008	Connection refused. Transport protocol error or negotiation failed.
0009	Open Failure in LC - Dialog protocol error or negotiation failed
000A	Open Failure in LC - Presentation protocol error or negotiation failed
000B	Open Failure in LC / Connection refused lack of system resources
000C	Open Failure in LC / Connection refused from GCOS7 duplicate user
000D	Open Failure in LC, Duplicate implicit LID / Q class not started
000E	Open Failure in LC, Duplicate GRTS Id / lack of memory resources
000F	Open Failure in LC, No Logical line declared for DACQ / 7 connection refused
0010	Open Failure in LC, GCOS 8 GW Missing translation / Incorrect device length in ILCRL.
0011	Open Failure in LC, DAC connection not initialized / Too many jobs executing
0012	Open Failure in LC, No binary transfer / impossible to start the IOF job

0013	Open Failure in LC, connection is not negotiated in FD mode / impossible to start the IOF job
0014	Disconnection - Timeout resulting from absence of traffic.
0016	Option missing for an RBF mailbox.
0017	Connection refused - Incorrect access right for MB.
0018	Connection refused - Incorrect access rights for the application.
0019	Connection refused - Unknown pre-negotiated message path
001A	Connection refused - Security validation failed.
001B	Connection refused - Unknown acceptor mailbox extension.
001C	Connection refused - Inoperable acceptor mailbox extension.
001D	Connection refused - Invalid Message group number.
001F	Disconnection - no more memory space.
0020	Connection refused - Unknown node.
0021	Connection refused - inaccessible node or Host down.
0022	Connection refused - saturated site.
0023	Connection refused - inoperable mailbox.
0024	(X.25) Packet too long. Problem with packet size. / Connection block already used.
0030	Syntax Error - option not known (received on close VC).
0031	(X.25) No response to call request packet - timer expired.
0033	(X.25) Timer expired for reset or clear indication.
0039	Disconnection - transport protocol error (MUX).
003C	Presentation Control Protocol Error
003E	The application has not the turn
003F	Message group closed
0040	(X.25) Facility code not allowed. / Connection refused - unknown node
0041	Connection refused - path not available.
0042	Connection refused - Duplicate USER ID / Facility parameter not allowed
0044	(X.25) Invalid calling address.
0045	(X.25) Invalid facility length.
0047	(X.25) No logical channel available.
004F	DNSC: (X.25) Invalid call packet length.
0050	Normal disconnection (GCOS3/8)
0051	Error or Event on LC initiated by GW
0052	Error or Event on LC initiated by GW.
0053	Error or Event on LC initiated by GW. TCall
0054	Error or Event on LC initiated by GW. DIA in LOCK State
0055	Error or Event on LC initiated by GW. DIA error

0056	Error or Event on LC initiated by GW. GW has no known explanation.
0057	Error or Event on LC initiated by GW. Reject mailbox permanent
0058	Error or Event on LC initiated by GW. No more input lines in DACQ
0059	Time-out on GCOS 3/8 gateway.
005A	Error or Event on LC initiated by GW. Disconnect from terminal without reason
005B	Error or Event on LC initiated by GW. Wrong letter or wrong record
005C	Error or Event on LC initiated by GW. Forbidden letter received
005D	Error or Event on LC initiated by GW. Forbidden letter received
005E	Error or Event on LC initiated by GW. No buffer for secondary letter
005F	Error or Event on LC initiated by GW. No buffer for fragmented letter
0060	Error or Event on LC initiated by GW. Disconnect on end of phase record
0061	Error or event on LC initiated by GW. No buffer for control letter.
0062	Error or event on LC initiated by GW. Mailbox in closing phase
0064	Error or event on LC initiated by GW. Flow control error.
0065	Error or event on LC initiated by GW. CH locked by operator.
0066	Error or event on LC initiated by GW. Disconnect with a normal TMG F2 exchange.
0067	Error or event on LC initiated by GW. Teletel rerouting error from DACQ
0068	Error or event on LC initiated by GW. Teletel routing error from DACQ
0069	Error or event on LC initiated by GW. Teletel rerouting error from TM
006A	Error or event on LC initiated by GW. Teletel rerouting error from TM
006B	Syntax error - text too long.
006C	Syntax error - illegal object in a GA command.
006D	Syntax error - unknown node Id.
0078	Syntax error - illegal command for this object.
0079	Syntax error - illegal date.
007F	(X.25) No route available for X.25 switching.
0081	No more network routes available for switching.
0082	(X.25) Hop count reached for X.25 switching.
0083	(X.25) Flow control negotiation error.
0085	(X.25) Frame level disconnection.
0086	(X.25) Frame level connection.

0087	(X.25) Frame level reset.
0090	Frame level not set.
0092	(X.25) X.25 Echo service in use.
0093	(X.25) Incorrect password for PAD connection.
0094	(X.25) No more PAD connections allowed.
0096	(X.25) TS SX25 or NU X25 objects locked.
009C	(X.25) Invalid packet header. X.25 protocol error.
009D	(X.25) Incompatible header. X.25 protocol error.
009E	(X.25) Logical Channel Number too high.
009F	(X.25) Incorrect packet type.
00B2	Use of invalid password through PAD
00B6	Unknown mailbox selection for PAD connection using the PAD password.
00C0	(X.25) Normal disconnection.
00D7	(X.25) TS image (of type DSA or DIWS) in LOCK state.
00DE	(X.25) NS RMT or NR SW in LOCK state.
00E1	Connection refused. Mailbox is not in ENBL state.
00E6	QOS not available permanently.
01xx	**Session Control**
0100	Logical connection accepted or normal termination
0101	Rejection for unknown reason or abnormal termination
0102	Acceptor node inoperable.
0103	Acceptor node saturated. When a node has no available resources
0104	Acceptor mailbox unknown.
0105	Acceptor mailbox inoperable.
0106	DNS: Acceptor mailbox saturated.
0107	DNS: Acceptor application program saturated.
0108	Transport protocol error or negotiation failed (DSA 200 only).
0109	Dialog protocol error or negotiation failed. (Wrong logical record).
010A	Time-out on session initiation / unknown LID
010B	Acceptor mailbox extension unknown.
010C	Acceptor mailbox extension inoperable.
010D	Invalid Session Number.
010E	Unknown node.
010F	System error. System generation error or insufficient memory space
0110	Application abnormal termination. Subsequent to an abnormal occurrence in the dialogue
0111	Normal terminate rejected.
0112	Protocol not supported.
0113	Session control service purged by user.
0115	Disconnection Time-out on message group initiation.

0117	Incorrect Access Right for MB
0118	Incorrect Access Right for the Application
0119	Pre-negotiated Message Path Descriptor unknown
011A	Security validation failed
011E	Incorrect object status
011F	Not enough memory space available.
0120	Node unknown.
0121	The channel object (CH) is in LOCK state
0122	Saturation - no plug available
0123	Object status = LOCK
0124	Connection block (TSCNX) already used
0125	Disconnection already running
0126	The connection block (TSCNX) is disconnected (or not connected)
0127	Change Credit value < 0
0128	Ineffective Change Credit (delta = 0)
0129	No more deferred letters
012B	"Reinitialization" Request
012C	"Reinitialization" in progress
012D	"Reinitialization" in progress, letters are dropped
012E	Close virtual circuit. Either no mapping exists between PA/NR or CL and VC/NS
012F	Null connection object index.
0130	Undefined function at Sysgen time.
0131	Letter too large with respect to the negotiated size.
0132	The received letter is longer than the size which was
0133	Disconnection of the session control user
0134	Interface error on EOR (End-Of-Record) processing.
013C	Presentation control protocol error.
013E	You do not have the turn.
013F	Message group closed.
0140	Session is closed.
0151	Request refused, no system buffers available.
0152	Incorrect addressing record.
0153	No presentation record in the ILCAL or ILCRL
0154	Negotiation failed on session mode
0156	Negotiation failed on resynchronization.
0157	Negotiation failed on END to END ACK
0158	No presentation record in the connection letter
0159	Negotiation failed on session mode
015A	Negotiation failed on letter size (in the Logical Connection record).

015B	Negotiation failed on resynchronization (in the Logical Connection record).
015C	Negotiation failed on end-to-end ACK (Logical Connection record).
015D	No support of the "letter" interface because Multirecord is not negotiated.
0160	Incorrect TSPACNX table.
0161	Protocol error on letter reception.
0162	Negotiation failure.
0163	Record header length error.
0164	Protocol error.
0165	Protocol error reception of control letter.
0166	Type or length error on interrupt letter.
0167	Protocol error on reception of data letter.
0168	Dialog protocol error.
0169	Unknown event.
016A	Protocol error on data transfer.
016B	Invalid status for a disconnection request.
016C	Invalid status for a recover
016D	Invalid status for a suspend/resume request.
016E	Negotiation failure.
016F	Unknown command.
0170	Error in presentation protocol
0171	Letter header length error in
0172	ILCAL is not DSA 200 protocol.
0173	Error in session record.
0174	Normal disconnection, without complementary reason code.
0175	Letter is not in ASCII or EBCD.
0176	Connection protocol letter header
0177	Letter header protocol error.
0178	Record header protocol error.
0179	Record header length error.
017A	Mbx record header length error.
017B	Error on buffer transfer.
017C	DSA 200 record header protocol
017D	DSA 300 record header protocol
017E	Unsupported connection options.
017F	Character error in ASCII string.
0180	No segmented record size.
0181	Invalid mailbox object index.
0182	Mapping error for a remote connection.
0190	No more buffers.

0191	Byte count is greater than GP.
0192	Byte count is greater than GP.
0193	Byte count is greater than GP.
0194	Byte count is greater than GP.
0195	Byte count is greater than GP.
0196	Byte count is greater than GP.
0197	Byte count is greater than GP.
0198	No more buffers.
0199	Byte count is greater than GP.
019A	Byte count is greater than GP.
019B	Byte count is greater than GP.
019C	Byte count is greater than GP.
019D	Byte count is greater than GP.
019E	Byte count is greater than GP.
019F	Byte count is greater than GP.
01A0	Invalid transfer state.
01A1	Suspend protocol running.
01A2	Suspend protocol running.
01A3	Recover protocol running.
01A4	Forbidden function in write request. ($WRITE)
01A5	Conflicting parameters for segmented record. (SWBREC)
01A6	Protocol conflict - suspend/recover.
01A7	Protocol not supported - letter/end-to-end ACK. (SWBLET)
01A8	Multi-record letter in progress.
01A9	Interrupt request forbidden.
01AA	Send control record request forbidden. (SCTROL)
01AB	Forbidden for TWA session - turn is here. (SREAD)
01AC	Termination forbidden - suspend or recover in progress. (STERM)
01C0	No space available for downstream connection request. (SMECNX)
01C1	No space available for upstream connection request. (SMUCNX)
01C2	No space available for upstream SCF connection. (SMRCNX)
01C3	No space available for session context. ($SCTX)
01E0	Enclosure or data length error for a write request. ($WRITE)
01E1	Enclosure or data length error for a write segment record request. (SWBREC)
01E2	Enclosure error for 'give turn' request. (SGVTRN)
01E3	Interrupt request is not demand turn, attention/data attention, or purge record.
01E4	Input status for a send control letter is not permitted.
01E8	Write request without turn.
01E9	Write segmented record request without turn.

01EA	Write segmented letter request without turn.
01EB	Send control letter request without turn.
01EC	Disconnection request without turn.
02xx	**Presentation Control**
0201	Protocol level not supported
0202	Application designation protocol error.
0203	Character encoding error. TM cannot support the proposed encoding.
0204	Character set error. TM cannot support the proposed character set.
0205	Character subset error. TM cannot support the proposed character subset.
0206	Incorrect record encoding.
0207	Incorrect parameter encoding.
0230	Data presentation control error. The presentation control proposed for this session cannot be used
0231	Device type is incompatible with the configuration.
0232	TM control protocol is incorrect.
0233	Device-sharing attributes are invalid.
0234	Initiator or acceptor configuration is not correct.
0235	Logical device index error.
0236	Number of logical devices is incompatible with the configuration.
0237	TM protocol record not supported.
03xx	**Terminal Management**
0300	Sysgen error WARNING. There is no mapped object; some objects will be spare.
0301	Operator requested session abort or logged.
0302	Idle time run out after secondary network failure.
0303	Idle time run out for no traffic.
0304	Form not found.
0305	Operator requested suspension.
0306	Destructive attention send on the session.
0307	Unknown TX addressed in this session. TM is unable to a the session.
030A	Protocol error. A record was received which did not comply with current standards
0310	Insufficient resources. The receiver cannot act on the request because of a temporary
031E	Incorrect value for Retry or Wait parameters on UP LL command.
0320	Function not supported.
0321	Parameter error. This can result
0322	Resource not available. The

0323	Intervention required (on principal device).
0324	Request not executable.
0325	EOI required.
0326	Presentation space altered, request executed.
0327	Presentation space altered, request not executed.
0328	Presentation space integrity lost.
0329	Device busy. The device is busy and cannot execute the request.
032A	Device disconnected.
032B	Resource not configured.
032C	Symbol set not loaded.
032D	Read partition state error.
032E	Page overflow.
0330	Subsidiary device temporarily not available.
0331	Intervention required at subsidiary device.
0332	Request not executable because of subsidiary device.
0340	TM cannot accept a new connection.
0341	Object status incorrect.
0342	The TM configuration is not correct.
0343	Unknown TX addressed on this session.
0344	Data presentation protocol error.
0345	Device type is incompatible with the configuration, or is not supported.
0346	TM control protocol incorrect.
0347	Device shareability attributes are invalid.
0348	Initiator or acceptor configuration is not correct.
0349	Logical device index error.
034A	Number of logical devices incompatible with the configuration.
0350	Disconnection of TM after reinitialization of the network.
0360	File not found. (Welcome and Broadcast Messages)
0361	Site not found. (Welcome and Broadcast Messages)
0362	NASF error. (Welcome and Broadcast Messages)
0370	No-session timeout. Device disconnected.
0371	No-input timeout. Device disconnected.
0372	No-output timeout. Device disconnected.
0373	Timeout due to no backup session being initiated.
0374	Timeout due to no backup session being established.
0375	Connection refused because of late activation of back up session.
0376	Disconnection of current session to switch to backup session.
0380	AUTOCN parameter not declared.
0381	Mixed ETB in data sent by VIP screen and cassette
0382	Data header sent by the terminal incorrect.

0383	Desynchronization in the exchange of data.
0384	KDS block count error.
038C	Remote terminal is not connected
0390	Unknown mailbox.
0391	No call packet to return.
0392	No "Possibility" command to return Protocol error
03C0	Slave device disconnection.
17xx	**Network Layer**
1701	PAD connection refused.
1702	Flow control error.
1706	Logical channel number not zero in restart packet.
1707	Illegal packet length or use of D-bit forbidden.
1708	Illegal header.
1709	Illegal Logical Channel Number.
1710	Invalid packet type for the automaton state. Protocol error
1711	Incorrect packet type.
1712	Inconsistent network parameters in the generation file.
1713	No more space.
1714	DSAC network layer object not usable.
1717	USED/ENBL transition. Transport station is locked.
1718	USED/ENBL transition. This is a back-up NR.
1719	USED/ENBL transition. Dynamic close due to load.
171A	USED/ENBL transition. Transfer time-out has elapsed.
171B	USED/ENBL transition. This is a back-up NR.
171C	USED/ENBL transition. Transport station is idle.
171E	USED/ENBL transition. NR object is locked.
171F	ENBL/LOCK transition. NR HDLC has no more memory space.
1721	Remote station is inaccessible via the configured network. Check
1723	Incorrect PAD password.
1724	Virtual circuit already in use. LCN (Logical Channel Number) too high.
1725	Invalid virtual circuit.
1726	Packet too short. Protocol error for the equipment directly connected to the Bull Datanet.
1727	Incompatibility between the generation parameters of two communicating systems on window or packet size.
1729	Packet size in communicating systems not the same.
1731	Timer runs out while waiting for call confirmation.
1732	Timer runs out while waiting for clear confirmation.
1733	Timer has run out while waiting a reset confirm.
1740	Call setup or call clearing problem.

1741	Open failure on virtual circuit. No flow control on this NS.
1742	Incorrect facility. Protocol error for the equipment directly connected to the Bull Datanet.
1744	Unknown subscriber.
1745	End of time-out on reset confirm. Invalid facility length. Protocol error for the equipment directly
1747	No logical channel available.
1749	End of time-out on call confirm.
174F	Incorrect packet length. Protocol error for the equipment directly connected to the Bull Datanet.
1755	Flow control, window, packet size or reset error.
1760	Frame disconnection.
1770	Frame connection.
1771	Frame reset.
1781	No more network routes available for X.25 switching.
1782	Maximum of 15 switches have been used,
1783	Flow control negotiation error.
1785	Frame level disconnection.
1786	Frame level connection.
1787	Frame level reset.
1790	Frame level not established.
1791	No more logical paths available for the PAD.
1792	Echo service busy.
1793	Incorrect PAD password.
1794	All the PAD virtual circuits are used
1795	X.25 initialization not possible.
179B	LCN not null in restart packet
179D	Incompatible header (receive error: all VC of concerned NS
179E	LCN greater than NBVC in NS directive
179F	Incorrect packet type
17A0	Invalid facility.
17B0	Normal disconnection.
17B1	X.25 Echo in use.
17B2	No more logical channels available.
17B3	No more PAD connections allowed.
17B4	TS SX25 or NU X25 object locked.
17B5	Buffer capacity overflow.
17B6	Normal disconnection.
17B8	Unknown calling SNPA (Sub-Network Point of Attachment).
17B9	Internet problem.
17CB	Call collision on VC

17CC	Incompatible generations (NR object without mapping).
17CE	Invalid status NR locked.
17CF	Lack of space.
17D0	Unknown subscriber.
17D4	TSCNX already used for another connection. SCF internal error.
17D7	Transport station locked.
17DD	Proper NS locked.
17DE	Invalid status NR locked.
17DF	Lack of space.
17E0	Forbidden parameter or invalid value.
17E1	Invalid transition.
17E2	Upward-mapped object (TS) not locked.
17E3	No object mapped above.
17E4	NR not locked (MP NR -ADD/-SUB) or virtual circuit already open.
17E5	NR is last in list and the TS is not locked.
17E6	No object mapped above (UP NR -PRIO). NR not mapped on TS.
17E7	Upward mapped object not locked
17E9	Mix of datagram and connection network
17EB	Class inconsistent with NR.
17EE	Incompatible generations. NR object without mapping.
17FF	Wrong parameter in administrative CALL
18xx	**Transport Layer**
1800	Normal disconnection initiated by the correspondent
1801	Local saturation at connection request time.
1802	Failed negotiation at connection time.
1803	Duplicate connection. Two or more requests have been issued for the same connection.
1804	Redundant request.
1805	Retransmission Time-out at transport level.
1806	Survey time-out at transport level.
1807	Transport protocol error.
1808	Session Control specified is not available (inaccessible).
1809	Requested Session Control Id unknown by remote transport.
180A	Termination because of disconnection by administration.
180B	Session Control/Transport interface error.
180C	Connection request on non-sharable VC in case of ISO Transport. ISO: header or parameter length is invalid.
1817	Station in shut-down state.
181F	No memory space at connection time.
1821	Session Control inaccessible by configured session routes. ISO: Session entity not attached to TSAP.

1824	Collision between Close NC and Open TC.
182E	Remote station not configured.
182F	Resource saturation.
1831	ISO: No route for the called NSAP.
1832	ISO: Received NSAP addresses are wrong.
1833	Segmentation violation.
1834	ISO:QOS priority not available temporarily, due to a local condition (for example, lack of resources).
1835	ISO:QOS priority permanently unavailable locally (for example, due to an error in the system generation).
183A	ISO: Remote reason not specified.
183C	ISO: Remote transport entity congestion at connect request time.
1840	Server in terminating state. TC has been re-assigned on another NC.
18A1	An additional NC has been assigned to a TC.
18B0	NC has been re-assigned on another VC.
18EF	Disconnection at Transport level caused by reception of RESTART DSA during the transfer phase.

Windows Sockets error Codes

Below is a list of Windows Sockets return codes and the corresponding description.

Hex code	**Windows Sockets Access Error name**	**Description**
2714	WSAEINTR	The (blocking) call was cancelled via WSACancelBlockingCall()
2719	WSAEBADF	The socket descriptor is not valid.
271E	WSAEFAULT	An invalid argument was supplied to the Windows Sockets API.
2726	WSAEINVAL	An invalid call was made to the Windows Sockets API.
2728	WSAEMFILE	No more file descriptors are available.
2733	WSAEWOULDBLOCK	The socket is marked as non-blocking and no connections are present to be accepted.
2734	WSAEINPROGRESS	A blocking Windows Sockets call is in progress.

2735	WSAEALREADY	The asynchronous routine being cancelled has already completed.
2736	WSAENOTSOCK	The descriptor is not a socket.
2737	WSAEDESTADDRREQ	A destination address is required.
2738	WSAEMSGSIZE	The datagram was too large to fit into the specified buffer and was truncated.
2739	WSAEPROTOTYPE	The specified protocol is the wrong type for this socket.
273A	WSAENOPROTOOPT	The option is unknown or unsupported.
273B	WSAEPROTONOSUPPORT	The specified protocol is not supported.
273C	WSAESOCKTNOSUPPORT	The specified socket type is not supported in this address family.
273D	WSAEOPNOTSUPP	The referenced socket is not a type that supports connection-oriented service.
273E	WSAEPFNOSUPPORT	
273F	WSAEAFNOSUPPORT	The specified address family is not supported by this protocol.
2740	WSAEADDRINUSE	The specified address is already in use.
2741	WSAEADDRNOTAVAIL	The specified address is not available from the local machine.
2742	WSAENETDOWN	The Windows Sockets implementation has detected that the network subsystem has failed.
2743	WSAENETUNREACH	The network address can't be reached from this host. There is probably a problem in the way you have set up TCP/IP routing for your PC (most likely you have not defined a default router).
2744	WSAENETRESET	The connection must be reset because the Windows Sockets implementation dropped it.
2745	WSAECONNABORTED	The connection has been closed.
2746	WSAECONNRESET	
2747	WSAENOBUFS	Not enough buffers available, or too many connections.
2748	WSAEISCONN	The socket is already connected.

2749	WSAENOTCONN	The socket is not connected.
274A	WSAESHUTDOWN	The socket has been shutdown.
274B	WSAETOOMANYREFS	
274C	WSAETIMEDOUT	Attempt to connect timed out without establishing a connection.
274D	WSAECONNREFUSED	The attempt to connect was forcefully rejected. The service on the other side is not available.
274E	WSAELOOP	Too many symbolic links were encountered in translating the path name.
274F	WSAENAMETOOLONG	
2750	WSAEHOSTDOWN	The host machine is out of service.
2751	WSAEHOSTUNREACH	The host machine is unreachable.
2752	WSAENOTEMPTY	
2753	WSAEPROCLIM	
2754	WSAEUSERS	
2755	WSAEDQUOT	
2756	WSAESTALE	
2757	WSAEREMOTE	
276B	WSASYSNOTREADY	Indicates that the underlying network subsystem is not ready for network communication.
276C	WSAVERNOTSUPPORTED	The version of Windows Sockets API support requested is not provided by this particular Windows Sockets implementation.
276D	WSANOTINITIALISED	A successful WSAStartup() must occur before using this API.
2AF9	WSAHOST_NOT_FOUND	Authoritative answer host not found.
2AFA	WSATRY_AGAIN	Non-authoritative answer host not found, or SERVERFAIL.
2AFB	WSANO_RECOVERY	Non-recoverable errors, FORMERR, REFUSED, NOTIMP.
2AFC	WSANO_DATA	Valid name, no data record of requested type.